Egyptian Informatics Journal 24 (2023) 100420

Contents lists available at [ScienceDirect](http://www.ScienceDirect.com/)

Egyptian Informatics Journal

journal homepage: [www.sciencedirect.com](http://www.sciencedirect.com/)

Full Length Article

A hybridization of granular adaptive tabu search with path relinking for the multi-depot open vehicle routing problem

Wenhan Shao, Tuanyue Xiao, Zhouxing Su, Junwen Ding [∗](#_bookmark0), Zhipeng Lü

*School of Computer Science and Technology, Huazhong University of Science and Technology, 1037 Luoyu Road, Wuhan, 430074, China*

A R T I C L E I N F O A B S T R A C T

*Keywords:*

Multi-depot open vehicle routing problem Granular neighborhoods

Solution-based tabu search Adaptive neighborhood selection Path relinking

The multi-depot open vehicle routing problem (MDOVRP) differs from the classical VRP in that there is more than one depot and the vehicle does not need to return to a depot after serving the last customer. For solving this challenging problem, we propose a hybrid metaheuristic algorithm (GATS-PR) which integrates the granular adaptive tabu search with path relinking. The main contributions of this work consist of introducing a solution- based tabu search technique in granular tabu search, designing an adaptive neighborhood selection method for the large neighborhoods with 22 kinds of move types, and adopting path relinking with a new similarity definition to the MDOVRP for the first time. Computational results on 24 public instances demonstrate that GATS-PR outperforms the previous state-of-the-art algorithms in the literature. Specifically, GATS-PR improved and matched the previous best known results on 4 and 19 instances, respectively.

# Introduction

Since the vehicle routing problem (VRP) was proposed in [[1](#_bookmark46)], its variants have been introduced to model various practical applications, including but not limited to city logistics [[2](#_bookmark47)], online food ordering de- livery [[3](#_bookmark48)], multi-robot exploration [[4](#_bookmark49)], hazardous waste collection [[5](#_bookmark50)] and telecommunications [[6](#_bookmark54)]. As a VRP variant, the multi-depot open vehicle routing problem (MDOVRP) was first formulated and solved to distribute fresh meat in an area of the city of Athens [[7](#_bookmark57)]. However, it received little attention until the public MDOVRP instances originat- ing from the multi-depot VRP benchmarks were proposed [[8](#_bookmark58)]. Because the VRP and its variants are NP-hard [[9](#_bookmark60)], it is diﬃcult for an exact al- gorithm to obtain optimal solutions in a reasonable time on large-scale instances of the MDOVRP [[10](#_bookmark62)–[12](#_bookmark66)]. In order to solve the MDOVRP effec- tively and eﬃciently, we propose the hybridization of granular adaptive tabu search with path relinking (GATS-PR). The main contributions of this study include:

1. The solution-based tabu search instead of attribute-based tabu search is devised in our granular tabu search.
2. The adaptive neighborhood selection method is developed for the large composite neighborhoods with 22 kinds of move types.
3. Path relinking is introduced to solve the MDOVRP for the first time and a new similarity definition is designed for the MDOVRP to guide the path generation.
4. We validate the performance of GATS-PR on 24 public instances. Computational results indicate that GATS-PR outperforms the pre- vious state-of-the-art algorithms.

The remaining of the paper is organized as follows. In Section [2](#_bookmark1), related work about the MDOVRP is presented. The definition of the MDOVRP is given in Section [3](#_bookmark2). Section [4](#_bookmark4) describes the proposed GATS- PR in detail. In Section [5](#_bookmark22), parameter tuning, computational results and importance analysis of the proposed algorithmic components are re- ported. Conclusion is given in Section [6](#_bookmark44).

# Related work

In this section, we first review the state-of-the-art algorithms for the MDOVRP. Then, we introduce the application of granular tabu search and path relinking in the VRP related field, which are important com- ponents of our proposed algorithm.

It is common to solve a problem by proposing its mixed-integer programming model and solving the model with commercial solvers [[13](#_bookmark68)–[15](#_bookmark70)]. This method is also widely used to solve the MDOVRP. In [[8](#_bookmark58)],

\* Corresponding author.

*E-mail addresses:* [shaowenhan95@hust.edu.cn](mailto:shaowenhan95@hust.edu.cn) (W. Shao), [ty\_xiaohust@hust.edu.cn](mailto:ty_xiaohust@hust.edu.cn) (T. Xiao), [suzhouxing@hust.edu.cn](mailto:suzhouxing@hust.edu.cn) (Z. Su), [junwending@hust.edu.cn](mailto:junwending@hust.edu.cn) (J. Ding), [zhipeng.lv@hust.edu.cn](mailto:zhipeng.lv@hust.edu.cn) (Z. Lü).

<https://doi.org/10.1016/j.eij.2023.100420>

Received 31 August 2023; Received in revised form 8 November 2023; Accepted 14 November 2023

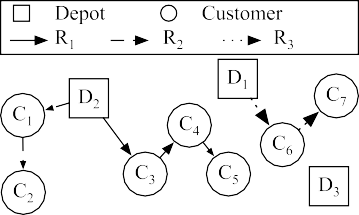
Available online 24 November 2023

1110-8665/© 2023 THE AUTHORS. Published by Elsevier BV on behalf of Faculty of Computers and Artificial Intelligence, Cairo University. This is an open access article under the CC BY-NC-ND license ([http://creativecommons.org/licenses/by-nc-nd/4.0/).](http://creativecommons.org/licenses/by-nc-nd/4.0/))

a mixed integer programming (MIP) formulation of the MDOVRP was proposed and solved with CPLEX. In [[11](#_bookmark64)] and [[16](#_bookmark73)], subtour elimina- tion constraints of [[8](#_bookmark58)] were improved and constraints related to the minimum vehicle number and the first customer to visit were added. In [[10](#_bookmark62)], a two-index vehicle flow formulation was proposed to solve the open location routing problem and the MDOVRP. In [[12](#_bookmark66)], the MIP model designed for the multi-depot open location routing problem with a heterogeneous fixed-charge fleet was adapted for the MDOVRP and solved with CPLEX.

In order to eﬃciently obtain high-quality solutions for combinato- rial optimization problems, numerous metaheuristic algorithms have been proposed, such as tabu search [[17](#_bookmark74)], evolutionary algorithms [[18](#_bookmark77)–[20](#_bookmark51)], hybrid genetic algorithms [[21](#_bookmark52)–[24](#_bookmark53)]. They are also applied for the MDOVRP. A hybrid genetic algorithm was proposed in [[8](#_bookmark58)], where a solution is encoded as a giant tour of all routes without trip delim- iters, the classic order crossover [[25](#_bookmark55)] is used to generate a new child solution from the chosen parent solutions, and three move types includ- ing 1-0 Exchange, 1-1 Exchange and 2-Opt are applied to improve the child solution. Tabu search with multiple neighborhood search were hy- bridized to solve the MDOVRP in [[26](#_bookmark56)], where multiple neighborhoods are generated by path moves and ejection chains. A memory-based it-

erated local search algorithm was presented for the MDOVRP in [[27](#_bookmark59)],



**Fig. 1.** An example of the solution of an MDOVRP instance.

# Problem definition

The MDOVRP is defined on an undirected complete graph *𝐺* = (*𝑉 , 𝐸*), where *𝑉* is the set of nodes and *𝐸* denotes the set of edges. *𝑉* consists of two exclusive subsets *𝑉 𝐷* and *𝑉 𝐶* which include |*𝑉 𝐷*| depots and |*𝑉 𝐶* | customers, respectively. A fleet with an unlimited number of homogeneous vehicles with capacity *𝑄* is based on each depot. Each customer *𝑣𝑖* ∈ *𝑉 𝐶* is characterized by a positive demand *𝑞𝑖* (*𝑞𝑖* ≤ *𝑄*). Each edge *𝑒𝑖𝑗* ∈ *𝐸, 𝑖* ≠ *𝑗* is associated with a positive cost *𝑐𝑖𝑗* represent- ing the traveling time from node *𝑣𝑖* to node *𝑣𝑗* . The costs are symmetric

and satisfy the triangular inequality.

A route *𝑅* = {*𝑣*0 *, 𝑣*1 *,* … *, 𝑣𝑚𝑅* } consists of *𝑚* different customers and

*𝑅 𝑅 𝑅 𝑅*

which records moves performed during the local search phase and uses

depot *𝑣*0 , where *𝑚𝑅* is the number of customers in *𝑅*. The traveling time

historical search information to guide the perturbation procedure. A variable tabu neighborhood search algorithm was proposed in [[28](#_bookmark61)],

*𝑅*

*𝑇𝑅* of *𝑅* is equal to

∑*𝑚𝑅* −1

*𝑖*=0

*𝑐𝑣𝑖 ,𝑣𝑖*+1 . The load *𝐿𝑅* is the total demand of

*𝑅*∑*𝑅*

all customers of *𝑅*, i.e., *𝐿* = *𝑚𝑅 𝑞 𝑖* . If *𝐿* ≤ *𝑄*, *𝑅* is a feasible route.

which applies granular local search in the intensification phase and the tabu shaking mechanism in the diversification phase. A multi-start metaheuristic algorithm was designed in [[12](#_bookmark66)], which consists of a con- structive heuristic and an iterated local search algorithm to solve the MDOVRP. In [[29](#_bookmark63)], a tabu search algorithm was designed for the simul- taneous scheduling of multi-project construction and vehicle routing, and it was adapted to the MDOVRP.

*𝑅*

Granular neighborhoods [[30](#_bookmark65)] and tabu search [[31](#_bookmark67)] are common in the field of the VRP, which are often adopted together to eﬃciently obtain high-quality solutions [[32](#_bookmark69)–[34](#_bookmark71)]. By excluding unsuitable moves in the current search phase, granular neighborhoods can improve the search eﬃciency without sacrificing the solution quality. To leverage the characteristic of granular neighborhoods, we design different granu- lar neighborhoods for each move type. Although the review shows that many studies have used tabu search to solve the MDOVRP and other VRP-related problems, all these studies choose the attribute-based tabu search strategy. Because it is diﬃcult to rapidly calculate hash values of solutions and quickly identify whether two solutions are identical, the solution-based tabu search strategy is rarely applied to the VRP and its variants. These challenges are addressed for the MDOVRP by designing the suitable hash function, adopting open addressing hashing with dou- ble hashing to handle hash collisions, and recording the characteristics of the visited solution rather than the solution itself.

Due to the diﬃculty in defining the similarity between solutions, there are few studies applying path relinking in the field of the VRP. In [[35](#_bookmark72)], the number of the same customers that appear in matched routes is defined as the similarity of two solutions for the VRP. Because the sequence of nodes on a route is ignored, the solution at the end of the path may not be the same as the guiding solution. Hence, the child solution may not be able to inherit good attributes of the guiding so- lution. In [[36](#_bookmark75)], the minimum number of relocating moves needed to transform one solution to another is defined as their similarity for the VRP. Therefore, the final solution is the same as the guiding solution af- ter path relinking. However, the diversity of the child solution is limited because only one move type is used. To overcome these disadvantages and take advantage of the characteristic that there is more than one de- pot in the MDOVRP, we design a new similarity definition and consider all the proposed move types at each step of the path generation.

*𝑅 𝑖*=1 *𝑣 𝑅*

The goal of the MDOVRP is to determine a set of feasible routes to serve

each customer exactly once with the minimum total traveling time of all routes.

Fig. [1](#_bookmark3) depicts an example of the solution of an MDOVRP instance, where seven customers are served by three routes beginning from de-

pots *𝐷*1 and *𝐷*2 , while depot *𝐷*3 is not used.

# The hybridization of granular adaptive tabu search with path relinking for the MDOVRP

To explore promising solution space systematically, we propose the GATS-PR algorithm which hybridizes granular adaptive tabu search

(GATS) with path relinking. GATS-PR manages population *𝑃* with *𝜇*

lution *𝑐ℎ𝑖𝑙𝑑* is generated by path relinking from two parent solutions feasible solutions as shown in Algorithm [1](#_bookmark5). At each iteration, a new so- randomly chosen from *𝑃* (lines 4-5) and is improved by the local search procedure GATS (line 6). If *𝑐ℎ𝑖𝑙𝑑* is feasible and does not belong to *𝑃* , its objective function value *𝑓* (*𝑐ℎ𝑖𝑙𝑑*) is compared with that of the worst parent solution *𝑝𝑤𝑜𝑟𝑠𝑡* (lines 7-8). If *𝑐ℎ𝑖𝑙𝑑* is better than *𝑝𝑤𝑜𝑟𝑠𝑡* , it re- places *𝑝𝑤𝑜𝑟𝑠𝑡* (lines 9-10). If *𝑐ℎ𝑖𝑙𝑑* is better than the best solution *𝑠*∗ found so far, it replaces *𝑠*∗ (lines 11-12).

**Algorithm 1** General framework of GATS-PR.

**Input:** Instance

**Output:** Best solution *𝑠*∗ found so far

1: Initialize population *𝑃* with *𝜇* feasible solutions (Section [4.2](#_bookmark23)) 2: *𝑠*∗ ← argmin*𝑠*∈*𝑃 𝑓* (*𝑠*)

3: **while** the time limit is not reached **do**

4: *𝑝*1 *, 𝑝*2 ← Randomly choose two solutions from *𝑃*

5: *𝑐ℎ𝑖𝑙𝑑* ← Apply path relinking to *𝑝*1*, 𝑝*2 (Section [4.3](#_bookmark16)) 6: *𝑐ℎ𝑖𝑙𝑑* ← Apply GATS to *𝑐ℎ𝑖𝑙𝑑* (Section [4.1](#_bookmark6))

7: **if** *𝑐ℎ𝑖𝑙𝑑* is feasible and *𝑐ℎ𝑖𝑙𝑑* ∉ *𝑃* **then**

8: *𝑝𝑤𝑜𝑟𝑠𝑡* ← argmax*𝑠*∈{*𝑝*1 *,𝑝*2 } *𝑓* (*𝑠*)

9: **if** *𝑓* (*𝑐ℎ𝑖𝑙𝑑*) *< 𝑓*(*𝑝𝑤𝑜𝑟𝑠𝑡* ) **then**

10: *𝑝𝑤𝑜𝑟𝑠𝑡* ← *𝑐ℎ𝑖𝑙𝑑*

11: **if** *𝑓* (*𝑐ℎ𝑖𝑙𝑑*) *< 𝑓*(*𝑠*∗) **then**

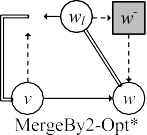
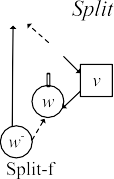
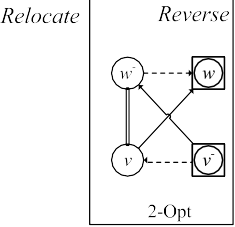
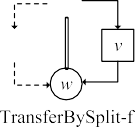
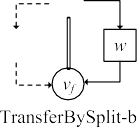
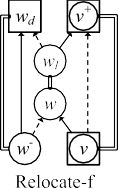
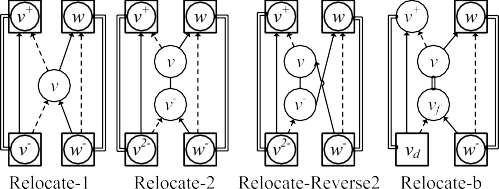
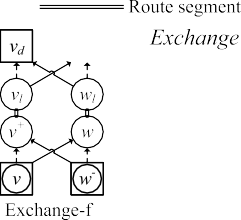
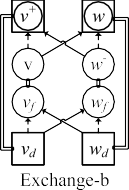
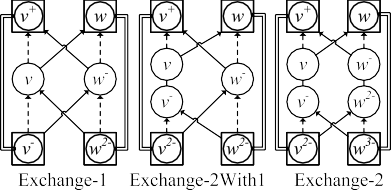
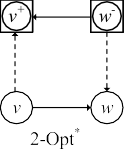
12: *𝑠*∗ ← *𝑐ℎ𝑖𝑙𝑑*

13: **end if**

14: **end if**

15: **end if**

16: **end while**



**Fig. 2.** Move types used in GATS-PR.

* 1. *Granular adaptive tabu search*

Algorithm [2](#_bookmark11) presents the main framework of granular adaptive tabu search (GATS). GATS iteratively improves the initial feasible solution

*𝑠* until the termination condition is satisfied (line 2), where *𝑖𝑡𝑒𝑚𝑎𝑥* is

nate GATS. At each iteration, *𝛿* customers are randomly chosen from the maximum consecutive iterations without improvement to termi- all customers (line 3) and a set *𝑀* of move types is determined by the

neighborhood solutions of *𝑠* generated according to *𝑀* , *𝐶𝑢𝑠* and granu- adaptive neighborhood selection method (line 4). After evaluating all

*𝑠*, *𝜌*(*𝑠*) defined in Eq. ([2](#_bookmark9)) is a normalization factor for relating the order

penalty, *𝑝𝑐𝑎𝑝* is the factor for penalizing the violation of vehicle capacity of magnitude of the objective function value and the vehicle capacity constraints, and *𝑣𝑐𝑎𝑝*(*𝑠*) is the cumulative violation of vehicle capacity

of constraint violations timely, *𝑝𝑐𝑎𝑝* is updated at each iteration in the constraints of all the tours as shown in Eq. ([3](#_bookmark10)). To control the degree

same way as in [[37](#_bookmark76)].

*𝑓* ′(*𝑠*)= *𝑓* (*𝑠*)+ *𝜌*(*𝑠*)× *𝑝𝑐𝑎𝑝* × *𝑣𝑐𝑎𝑝*(*𝑠*) (1)

|*𝑉 𝐶* | × *𝑓* (*𝑠*)

lar neighborhoods of *𝐶𝑢𝑠*, we set *𝑠* to the best unvisited neighborhood

solution in terms of the generalized objective function value (line 5),

*𝜌*(*𝑠*)=

∑

*𝑖*∈*𝑉 𝐶 𝑞𝑖*

(2)

record *𝑠* as a tabu solution (line 6), and update the best solution *𝑠*∗

found so far (lines 7-11).

*𝑣𝑐𝑎𝑝*

∑

(*𝑠*)= *𝑚𝑎𝑥*(*𝐿𝑅*

*𝑅*∈*𝑠*

– *𝑄,* 0) (3)

**Algorithm 2** General framework of GATS.

**Input:** Initial feasible solution *𝑠* **Output:** Best solution *𝑠*∗ found so far 1: *𝑖𝑡𝑒𝑢𝑛𝐼 𝑚𝑝* ← 0*, 𝑠*∗ ← *𝑠*

2: **while** *𝑖𝑡𝑒𝑢𝑛𝐼 𝑚𝑝* ≤ *𝑖𝑡𝑒𝑚𝑎𝑥* **do**

3: *𝐶𝑢𝑠* ← Randomly choose *𝛿* customers

4: *𝑀* ← Choose move types with the adaptive neighborhood selection mechanism 5: *𝑠* ← Choose the best unvisited neighborhood solution of *𝑠*

6: Record *𝑠* as a tabu solution in hash tables 7: **if** *𝑠* is feasible and *𝑓* (*𝑠*) *< 𝑓*(*𝑠*∗) **then**

8: *𝑖𝑡𝑒𝑢𝑛𝐼 𝑚𝑝* ← 0*, 𝑠*∗ ← *𝑠*

9: **else**

10: *𝑖𝑡𝑒𝑢𝑛𝐼 𝑚𝑝* ← *𝑖𝑡𝑒𝑢𝑛𝐼 𝑚𝑝* +1

11: **end if**

12: **end while**

* + 1. *Generalized objective function*

In GATS, it is allowed to visit infeasible solutions which violate ve- hicle capacity constraints to explore wider search space. In order to

design a generalized objective function *𝑓* ′ for the MDOVRP as shown compare the quality of feasible solutions with that of infeasible ones, we in Eq. ([1](#_bookmark8)), where *𝑓* (*𝑠*) denotes the objective function value of solution

* + 1. *Move types*

Let *𝑣* and *𝑤* represent two nodes, *𝑣𝑘*− and *𝑣𝑘*+ stand for the *𝑘*th predecessor and successor nodes of *𝑣* (*𝑘* is omitted when *𝑘* = 1). When

*𝑣* is a customer, *𝑣𝑓* and *𝑣𝑙* denote the first and the last customers of the tour including *𝑣*, and *𝑣𝑑* represents the depot serving *𝑣*. A total of

22 move types in 6 categories are depicted in Fig. [2](#_bookmark7) and described as follows:

1. *𝐸𝑥𝑐ℎ𝑎𝑛𝑔𝑒* interchanges distinct customers. Exchanged customers

are served by the same depot but belong to different tours in 2-

Opt\* and by different depots in Exchange-b and Exchange-f. For Exchange-1, Exchange-2 and Exchange-2With1, there is no limita- tion for exchanged customers.

1. *𝑅𝑒𝑙𝑜𝑐𝑎𝑡𝑒* moves one or several consecutive customers into a new

reversed and it is unchanged for other *𝑅𝑒𝑙𝑜𝑐𝑎𝑡𝑒* move types. After position. For Relocate-Reverse2, the order of relocated customers is

*𝑅𝑒𝑙𝑜𝑐𝑎𝑡𝑒*, there must be at least one customer left on the affected

tours.

1. *𝑅𝑒𝑣𝑒𝑟𝑠𝑒* reverses a route segment and it includes 2-Opt.
2. *𝑆𝑝𝑙𝑖𝑡* extracts a route segment of a tour to construct a new tour

served by the same or a different depot. After the operation, there

is at least one customer left on the affected tours. Hence, the tour number is increased by one.

1. *𝑀𝑒𝑟𝑔𝑒* combines two tours into one, which reduces the tour num-

ber by one.

1. *𝑇𝑟𝑎𝑛𝑠𝑓 𝑒𝑟* changes the depot of a tour to another.

Although a vehicle does not return to a depot in the MDOVRP, we add an auxiliary edge linking the last customer with the departing depot to simplify the design of move types. The edge cost from a customer to a depot is set to zero.

lution. *𝐸𝑥𝑐ℎ𝑎𝑛𝑔𝑒* swaps customers, *𝑅𝑒𝑙𝑜𝑐𝑎𝑡𝑒* changes the position of These six categories of move types differ in how they modify a so- customers, *𝑅𝑒𝑣𝑒𝑟𝑠𝑒* changes the direction of a route segment, *𝑆𝑝𝑙𝑖𝑡* ex-

tracts a route segment from the original tour and serves it with a depot,

*𝑀𝑒𝑟𝑔𝑒* combines two tours into one, and *𝑇𝑟𝑎𝑛𝑠𝑓 𝑒𝑟* severs the tour with

another depot. Additionally, the impact degree on the solution varies

for distinct move types within each category. For example, relocating one customer has a smaller impact on the solution than relocating a lot of customers. By designing large composite neighborhoods with differ- ent functions and impact degrees, the algorithm can select appropriate move types in different search phases and achieve the balance between intensification and diversification of the search.

* + 1. *Adaptive neighborhood selection*

If evaluating neighborhood solutions generated by all the move types at each iteration, it will be too time-consuming. Hence, we de- velop an adaptive neighborhood selection (ANS) strategy to choose suitable move types and group all the move types into four neighbor-

hoods *𝑁* = {*𝑁*1*, 𝑁*2*, 𝑁*3*, 𝑁*4} in ascending order of the degree of change

to a solution as follows:

* + - * *𝑁*1 = {Exchange-1, Relocate-1, Relocate-2, RelocateWithReverse- 2, 2-Opt}
      * *𝑁*2 = {2-Opt\*, Exchange-2, Exchange2With1}
      * *𝑁*3 = {Exchange-b, Exchange-f, Relocate-b, Relocate-f, TransferBySplit-b, TransferBySplit-f}
      * *𝑁*4 = {MergeBy2-Opt\*, MergeByRelocate-b, MergeByRelocate-f, Split-0, Split-1, Split-2, Split-b, Split-f}

At each iteration, ANS selects neighborhood *𝑛𝑒𝑖* ∈ *𝑁* with probabil- ity *𝑝𝑛𝑒𝑖* as shown in Eq. ([4](#_bookmark13)), where *𝑤𝑛𝑒𝑖* is the weight of selecting *𝑛𝑒𝑖*.

To avoid too large weig[ht differenc]e among neighborhoods, weights are

intensification of the search because the neighborhoods whose general- ized objective function value improvements are sub-optimal are likely to be frequently selected. 2) If a move type is chosen at an unsuitable time, its weight will be set to a small value and it is hard to select it again when there are too many neighborhoods. This reduces diversifi- cation brought by various move types.

high for move types in *𝑁*1 and *𝑁*2 in the early stages of the search, these Due to the fact that the probabilities of improving the solution are

two neighborhoods have high weights and are frequently selected. Be- cause a lot of move types among these neighborhoods are compared to obtain the best improvement move, it guarantees intensification of the search. When the search falls into a local optimum, weights of selecting

*𝑁*1 and *𝑁*2 are decreased because move types in *𝑁*1 and *𝑁*2 are un-

of choosing *𝑁*3 and *𝑁*4, and diversifies the search at suitable time. likely to improve the solution quality. This increases the probabilities

* + 1. *Granular neighborhoods*

choosing a move type, a customer *𝑐𝑢𝑠*, and a neighboring node of *𝑐𝑢𝑠* As shown in Fig. [2](#_bookmark7), a neighborhood solution is determined when

orating solution quality, we define granular neighborhoods of *𝑐𝑢𝑠* after after the move. In order to further speed up the search without deteri- the move by considering only proximate nodes of *𝑐𝑢𝑠* to limit the num-

ber of neighborhood solutions as follows:

* + - * All depots.
      * The first ⌊*𝑔𝑡* × |*𝑉 𝐶* |⌋ closest customers of *𝑐𝑢𝑠*, where *𝑔𝑡* is the gran- ularity threshold and its value is between *𝑔𝑡𝑚𝑖𝑛* and *𝑔𝑡𝑚𝑎𝑥* .
      * Neighboring nodes of *𝑐𝑢𝑠* in the best solution found so far.

We update *𝑔𝑡* in the same way as mentioned in [[37](#_bookmark76)].

* + 1. *Solution-based tabu search*

For a population-based hybrid evolutionary algorithm, when the attribute-based tabu search strategy is used, a solution may be visited many times in the local search even if initial solutions are different. Thus, we design a solution-based tabu search strategy which records all visited solutions and avoids revisiting them to further enhance diversi- fication of the search.

Let *𝑙* represent the list with |*𝑉 𝐶* |+|*𝑉 𝐷*| random numbers and *𝑙𝑖* the

value of the *𝑖*th element of *𝑙*. The hash value *ℎ* of solution *𝑠* is calculated by the hash function *𝐻* as shown in Algorithm [3](#_bookmark12). For each edge of *𝑠*, corresponding values of its two ends in *𝑙* are multiplied to XOR with

limited to the interval

*𝑤𝑚𝑖𝑛, 𝑤𝑚𝑎𝑥*

and their initial values are *𝑤𝑚𝑖𝑛* . Let

*ℎ* (line 5), where *ℎ* is initially set to zero (line 1). The hash value of *𝑠*

*𝑠* denote the current solution, *𝑠*′ stand for the solution after the move, and *𝑠𝑒𝑙* represent the selected neighborhood. After the move, ANS up- dates weight *𝑤𝑠𝑒𝑙* as follows:

1. The accumulative generalized objective function value improve-

and inserted edge of the move, the corresponding values in *𝑙* of its two after a move is calculated as shown in Algorithm [4](#_bookmark17). For each deleted ends are multiplied to XOR with *ℎ* to obtain the hash value of *𝑠* after the

time complexity of Algorithm [4](#_bookmark17) is *𝑂*(1), which ensures the eﬃciency of move (lines 3 and 7). Due to the characteristics of the XOR operator, the

∗

ment Δ

*𝑠𝑒𝑙*

of *𝑠𝑒𝑙* is updated as shown in Eq. ([5](#_bookmark14)), where discount

the neighborhood evaluation.

factor *𝛼* is introduced to balance historical and current contribu-

tions of neighborhoods.

1. ANS adjusts *𝑤𝑠𝑒𝑙* by comparing Δ∗ with the median value

**Algorithm 3** Hash value of a solution.

**Input:** Solution *𝑠*

*𝑚𝑒𝑑*(Δ∗) of Δ∗

*𝑠𝑒𝑙*

as shown in Eq. ([6](#_bookmark15)), where factor

*𝜆* is used to con-

**Output:** Hash value *ℎ* of *𝑠*

trol the convergence speed of weights. If the former is less than the

latter, it indicates that *𝑠𝑒𝑙* is more likely to improve the solution.

1: *ℎ* ← 0

2: **for** *𝑡𝑜𝑢𝑟* ∈ *𝑠* **do**

*𝑝𝑛𝑒𝑖*

Therefore, *𝑤*

= ∑*𝑤𝑛𝑒𝑖*

*𝑤*

*𝑠𝑒𝑙*

is increased. Otherwise, *𝑤𝑠𝑒𝑙*

is reduced.

(4)

3: **for** *𝑒𝑑𝑔𝑒* ∈ *𝑡𝑜𝑢𝑟* **do**

4: *𝑛𝑜𝑑𝑒*1 *, 𝑛𝑜𝑑𝑒*2 ← Two nodes of *𝑒𝑑𝑔𝑒*

5: *ℎ* ← (*𝑙𝑛𝑜𝑑𝑒*1 × *𝑙𝑛𝑜𝑑𝑒*2 ) *⊕ℎ*

6: **end for**

*𝑖*∈*𝑁*

Δ∗ = *𝛼* × Δ∗

*𝑖*

+ (1− *𝛼*)× (*𝑓* ′(*𝑠*′)− *𝑓* ′(*𝑠*)) (5)

7: **end for**

*𝑠𝑒𝑙* {

*𝑤𝑠𝑒𝑙* =

*𝑠𝑒𝑙*

*𝑤𝑠𝑒𝑙* + *𝜆* × (*𝑤𝑚𝑎𝑥* − *𝑤𝑠𝑒𝑙* ) Δ∗

*𝑤𝑠𝑒𝑙* + *𝜆* × (*𝑤𝑚𝑖𝑛* − *𝑤𝑠𝑒𝑙* ) Δ∗

*𝑠𝑒𝑙*

*𝑠𝑒𝑙*

*< 𝑚𝑒𝑑*(Δ∗)

≥ *𝑚𝑒𝑑*(Δ∗)

(6)

To handle hash collisions, we adopt classic open addressing hash- ing with double hashing. In open addressing hashing, probing refers to obtaining offsets of the hash value calculated by the hash function

If we define a neighborhood for each move type, there will be two disadvantages: 1) Weights of many move types are comparable and the neighborhood selection is random for these move types. This reduces

*𝐻* when a collision occurs. Due to probing, open addressing hash-

ing is able to place items with the same hash value in different po-

sitions. Double hashing is one of the probing methods, which uses

**Algorithm 4** Hash value of a solution after a move.

**Input:** Hash value *ℎ* of the current solution, all deleted and inserted edges *𝐸𝑑* and *𝐸𝑖* of

1: **for** *𝑒𝑑𝑔𝑒* ∈ *𝐸𝑑* **do** the move, respectively

2: *𝑛𝑜𝑑𝑒*1 *, 𝑛𝑜𝑑𝑒*2 ← Two nodes of *𝑒𝑑𝑔𝑒*

3: *ℎ* ← *ℎ ⊕* (*𝑙𝑛𝑜𝑑𝑒*1 × *𝑙𝑛𝑜𝑑𝑒*2 )

4: **end for**

5: **for** *𝑒𝑑𝑔𝑒* ∈ *𝐸𝑖* **do**

6: *𝑛𝑜𝑑𝑒*1 *, 𝑛𝑜𝑑𝑒*2 ← Two nodes of *𝑒𝑑𝑔𝑒*

7: *ℎ* ← *ℎ ⊕* (*𝑙𝑛𝑜𝑑𝑒*1 × *𝑙𝑛𝑜𝑑𝑒*2 )

8: **end for**

the second hash function *𝑆𝐻* to calculate offsets. In our experiment,

*𝑆𝐻*(*𝑠*) = *𝐻* (*𝑠*) mod 97 + 1.

Let *𝑙𝑒𝑛* = 999983 represent the size of hash table *𝐻𝑇* , *𝑚* stand for

1. Solution construction: For each depot *𝑑*, customers assigned to *𝑑*

are randomly selected and inserted into a tour served by *𝑑* one at

a time according to the following rules:

* + The customer is inserted into a tour with the minimum cost if it does not violate vehicle capacity constraints;
  + Otherwise, the customer is served by a new tour.

*4.3. Path relinking for the MDOVRP*

Let *𝑆𝐷𝑖,𝑗* (*𝑆𝑁𝑖,𝑗* ) represent the number of customers served by the same depot (with the same neighbors) in solutions *𝑠𝑖* and *𝑠𝑗* . The simi- larity *𝑠𝑖𝑚𝑖,𝑗* between *𝑠𝑖* and *𝑠𝑗* is defined as the weighted sum of *𝑆𝐷𝑖,𝑗* and *𝑆𝑁𝑖,𝑗* as shown in Eq. ([9](#_bookmark18)), where *𝑤𝑆𝐷* is the weight factor of *𝑆𝐷𝑖,𝑗* .

the number of solutions in *𝐻𝑇* , and *𝑙𝑓* = *𝑚*∕*𝑙𝑒𝑛* denote the load fac- tor of *𝐻𝑇* . If solution *𝑠* is recorded in the *𝑖*th row of *𝐻𝑇* , *𝐻𝑇**𝑖* =

*𝑠𝑖𝑚𝑖,𝑗*

= *𝑤*

*𝑆𝐷*

× *𝑆𝐷𝑖,𝑗*

+ (1− *𝑤*

*𝑆𝐷*

)× *𝑆𝑁𝑖,𝑗*

(9)

(*𝑓* (*𝑠*)*, 𝑣𝑐𝑎𝑝*(*𝑠*)), where *𝑓* (*𝑠*) is the objective function value of solution

*𝑠* and *𝑣𝑐𝑎𝑝*(*𝑠*) is the cumulative violation of vehicle capacity constraints of *𝑠*. The initial mapped position *𝐼𝑃* (*𝑠*) of solution *𝑠* and the *𝑖*th mapped position *𝑃𝑖*(*𝑠*) after double hashing are shown in Eq. ([7](#_bookmark19)) and Eq. ([8](#_bookmark20)), re- spectively. When searching for an empty position to record solution *𝑠* or determining whether *𝑠* has existed in *𝐻𝑇* , we check values recorded in rows *𝑅* = {*𝐼𝑃* (*𝑠*)*, 𝑃*1(*𝑠*)*, 𝑃*2(*𝑠*)*,* …} of *𝐻𝑇* in turn until finding a row

*𝑟* ∈ *𝑅*, where *𝐻𝑇𝑟* is empty or equal to ((*𝑓* (*𝑠*)*, 𝑣𝑐𝑎𝑝* (*𝑠*)).

*𝐼𝑃* (*𝑠*)= *𝐻* (*𝑠*) mod *𝑙𝑒𝑛* (7)

*𝑃𝑖*(*𝑠*)= (*𝐻* (*𝑠*)− *𝑖* × *𝑆𝐻*(*𝑠*)) mod *𝑙𝑒𝑛, 𝑖* ∈ ℕ+ (8) For the current row *𝑟* ∈ *𝑅*:

* If *𝐻𝑇𝑟* is equal to (*𝑓* (*𝑠*)*, 𝑣𝑐𝑎𝑝*(*𝑠*)), we consider that solution *𝑠* has been visited;
* If *𝐻𝑇𝑟* is empty, it indicates that *𝑠* is an unvisited solution;
* Otherwise, the value in the next row of *𝐻𝑇* is checked.

When load factor *𝑙𝑓* is large, the search eﬃciency of the hash table will significantly decrease. Hence, when *𝑙𝑓 >* 0*.*75: An unvisited solu- tion *𝑠* will be recorded in *𝐻𝑇𝐼𝑃* (*𝑠*) if its objective function value *𝑓* (*𝑠*) is better than that recorded in *𝐻𝑇𝐼𝑃* (*𝑠*) ; Otherwise, *𝑠* is discarded.

acceptable computational cost, we use three hash tables *𝑇𝐵*1, *𝑇𝐵*2 and To avoid identifying an unvisited solution as a visited one under

*𝑇𝐵*3 with different values in *𝑙* to record solutions. Only when the pair of *𝑓* (*𝑠*) and *𝑣𝑐𝑎𝑝*(*𝑠*) exists in the mapped positions of all the tables, *𝑠* is

regarded as a tabu solution. Fig. [3](#_bookmark25) illustrates an example of determining

three customers 2, 3 and 4. Solution *𝑠* includes one tour departing from whether a solution is in tabu state. There are two depots 0 and 1 and depot 0, visiting customers 2, 3 and 4 in turn. The pair *𝑝* of *𝑓* (*𝑠*) and

*𝑣𝑐𝑎𝑝*(*𝑠*) is (9, 2) for solution *𝑠*. For *𝐻𝑇* 1, *𝐼𝑃* (*𝑠*) = 22 and *𝐻𝑇* 1 is equal

tial solution *𝑝*1 and guiding solution *𝑝*2 , the similarity *𝑠𝑖𝑚𝑐𝑢𝑟* between Algorithm [5](#_bookmark21) presents the pseudo-code of our path relinking. For ini-

them is calculated (line 1). A high-quality solution is more likely to be obtained from solutions in the middle part of the path after applying the GATS procedure because it inherits good attributes of parent solutions.

Consequently, parameters *𝑟𝑎𝑡𝑖𝑜𝑏𝑒𝑔* and *𝑟𝑎𝑡𝑖𝑜𝑒𝑛𝑑* are used to limit the

interval of the path where the final result is selected (line 2). In the

ing vehicle capacity constraints too much and push *𝑝*1 toward *𝑝*2, only path construction (lines 3-12), to avoid the final solution from violat- feasible neighborhood solutions whose similarities to *𝑝*2 are larger than

*𝑠𝑖𝑚𝑐𝑢𝑟* are considered (lines 4-5). We sort solutions of *𝑁𝑒𝑖* in an as-

cending order of the sum of their rankings in terms of similarity and

dom solution from the first *𝛾* ones of *𝑁𝑒𝑖* as new *𝑝*1 (line 7), *𝑠𝑖𝑚𝑐𝑢𝑟* and objective function value improvements (line 6). After choosing a ran-

the reference solution are updated (lines 8-11).

**Algorithm 5** Path relinking for the MDOVRP.

**Input:** Initial solution *𝑝*1, guiding solution *𝑝*2

**Output:** Reference solution *𝑠𝑟𝑒𝑓*

1: *𝑠𝑖𝑚𝑐𝑢𝑟* ← Calculate similarity between *𝑝*1 and *𝑝*2

2: *𝑠𝑖𝑚𝑏𝑒𝑔* ← *𝑠𝑖𝑚𝑐𝑢𝑟* × *𝑟𝑎𝑡𝑖𝑜𝑏𝑒𝑔 , 𝑠𝑖𝑚𝑒𝑛𝑑* ← *𝑠𝑖𝑚𝑐𝑢𝑟* × *𝑟𝑎𝑡𝑖𝑜𝑒𝑛𝑑 , 𝑓* (*𝑠𝑟𝑒𝑓* ) ← +∞

3: **while** *𝑠𝑖𝑚𝑐𝑢𝑟* ≤ *𝑠𝑖𝑚𝑒𝑛𝑑* **do**

4: *𝑁𝑒𝑖* ← Generate all the feasible neighborhood solutions of *𝑝*1 with all the move

types defined in Fig. [2](#_bookmark7)

5: *𝑁𝑒𝑖* ← Retain *𝑠* ∈ *𝑁𝑒𝑖* whose similarity to *𝑝*2 is greater than *𝑠𝑖𝑚𝑡𝑚𝑝*

6: *𝑁𝑒𝑖* ← Sort *𝑁𝑒𝑖* according to the sum of rankings of *𝑠* in *𝑁𝑒𝑖* in terms of similarity

and objective function value improvements

7: *𝑝*1 ← Randomly pick a solution from the first *𝛾* ones of *𝑁𝑒𝑖*

8: *𝑠𝑖𝑚𝑐𝑢𝑟* ← Calculate similarity between *𝑝*1 and *𝑝*2

9: **if** *𝑠𝑖𝑚𝑐𝑢𝑟* ≥ *𝑠𝑖𝑚𝑏𝑒𝑔* and *𝑓* (*𝑠𝑟𝑒𝑓* ) *> 𝑓*(*𝑝*1 ) **then**

10: *𝑠𝑟𝑒𝑓* ← *𝑝*1

11: **end if**

12: **end while**

to *𝑝*. Therefore, solution *𝑠*

22

is considered as a visited solution in

*𝐻𝑇* 1.

The pairs of *𝐻𝑇* 2 and *𝐻𝑇* 3 are different from *𝑝*, so we check the first

# 5 40 5. Experimental results

mapped positions after double hashing in these tables, which are both

999982. Because *𝐻𝑇* 2 and *𝐻𝑇* 3 are both empty, solution *𝑠* is

999982

999982 2 3

We tested GATS-PR on the public MDOVRP instances introduced

regarded as an unvisited solution in *𝐻𝑇*

unvisited solution.

* 1. *Population initialization*

and *𝐻𝑇* . Hence, *𝑠* is an

in [[8](#_bookmark58)].[1](#_bookmark24) Table [1](#_bookmark26) presents the notations used in this section. In the fol- lowing subsections, we first present the parameter tuning process of GATS-PR. Then, the results obtained by GATS-PR are compared with those obtained by the state-of-the-art reference algorithms in the liter-

Population *𝑃* is initialized with *𝜇* feasible solutions, each of which

is generated after the following two steps and is improved by the local

search procedure GATS.

1. Customer assignment: Each customer *𝑐𝑢𝑠* is assigned to its nearest depot for 0*.*7 × *𝜇* solutions and randomly assigned to one of its two

nearest depots according to the probability 1 − *𝑐𝑑,𝑐𝑢𝑠* for the

ature. Finally, the importance of the proposed algorithmic components is analyzed.

* 1. *Parameter tuning*

For instances which are easy to solve, it is not sensitive to parame- ter settings. Hence, instances p09, p10 and pr05 which are hard to solve

∑

2

*𝑐 𝑖*

remaining 0*.*3 × *𝜇* solutions, where *𝑑𝑖*

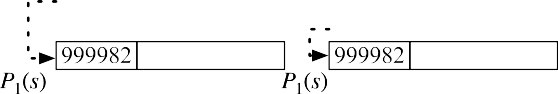
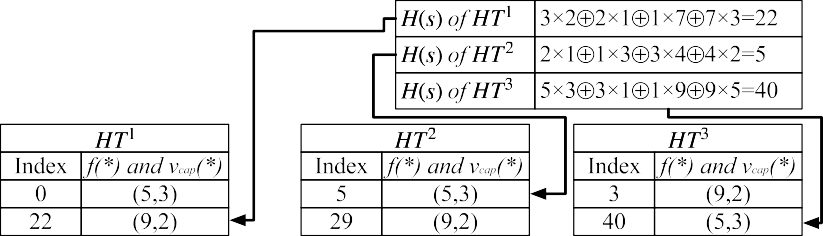
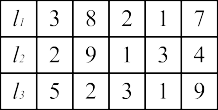
*𝑐𝑢𝑠*.

*𝑖*=1 *𝑑𝑐𝑢𝑠 ,𝑐𝑢𝑠*

is the *𝑖*th nearest depot of

*𝑐𝑢𝑠*

1 <http://neo.lcc.uma.es/vrp/vrp-instances/multiple-depot-vrp-instances/>.



**Fig. 3.** An example of determining whether a solution is in tabu state.

**Table 1**

Notations.

Notation Meaning

Ins Instance name

BKS Objective function value of the best known solution

Ref Reference to the algorithm which obtained BKS for the first time

*𝑓𝑏𝑒𝑠𝑡* Objective function value of the best solution

*𝑓𝑎𝑣𝑔* Average *𝑓𝑏𝑒𝑠𝑡* among multiple experiments

* The number *𝛾* of elite solutions from which the neighborhood move

is picked at each step of path generation. Its default value is 3.

* The weight factor *𝑤𝑆𝐷* of *𝑆𝐷𝑖,𝑗* . Its default value is 0.7.
* The combination of the population size *𝜇* and *𝑖𝑡𝑒𝑚𝑎𝑥* . Its default

value is (10, 2).

* The combination of *𝑟𝑎𝑡𝑖𝑜𝑏𝑒𝑔* and *𝑟𝑎𝑡𝑖𝑜𝑒𝑛𝑑* determining the path seg-

ment of interest. Its default value is (0.6, 0.4).

The maximal run time was 600 seconds for tuning path relinking-related

*𝐺𝑎𝑝*

*𝑏𝑒𝑠𝑡*

(*𝑓𝑏𝑒𝑠𝑡*

-BKS)/BKS reported as ‰ values

parameters. Based on *𝑆𝐺𝑏𝑒𝑠𝑡* presented in Table [3](#_bookmark29), *𝛾* was set to 5, *𝑤𝑆𝐷*

*𝐺𝑎𝑝*

*𝑎𝑣𝑔*

(*𝑓𝑎𝑣𝑔*

-BKS)/BKS reported as ‰ values

to 0.7, *𝜇* to 10, *𝑖𝑡𝑒𝑚𝑎𝑥* to 1, *𝑟𝑎𝑡𝑖𝑜𝑏𝑒𝑔* to 0.5 and *𝑟𝑎𝑡𝑖𝑜𝑒𝑛𝑑* to 0.2.

*𝑡𝑎𝑣𝑔* Average run time of obtaining the best solutions in seconds

*𝐴𝑣𝑒* Average *𝐺𝑎𝑝𝑏𝑒𝑠𝑡* , *𝐺𝑎𝑝𝑎𝑣𝑔* or *𝑡𝑎𝑣𝑔*

- Corresponding value is not given in the literature

were selected to tune parameters. Note that similar statistics can be ob- served on the remaining instances. Because the local search procedure GATS and the path relinking operator are relatively independent, their parameters were tuned by applying GATS and GATS-PR to these in- stances, respectively. For each parameter, a default value and possible values were given. When tuning a parameter, other parameters were fixed to default values to avoid explosions in the number of parame- ter combinations. We first tuned GATS-related parameters and set their default values to the selected values when tuning path relinking-related parameters. Three independent runs were performed for the selected in- stances. In order to normalize the results of different instances, the sum

*𝑆𝐺𝑏𝑒𝑠𝑡* of *𝐺𝑎𝑝𝑏𝑒𝑠𝑡* of each selected instance is introduced. The parame-

ter value whose corresponding *𝑆𝐺𝑏𝑒𝑠𝑡* is the minimum was selected as

* 1. *Comparison of GATS-PR with the state-of-the-art algorithms*

In Tables [4](#_bookmark30) and [5](#_bookmark31), the detailed experimental environment of the reference exact and metaheuristic algorithms are given, where column Name represents the name of the reference algorithms, column Num lists the number of independent runs, column CPU gives the compu- tational configuration, and column Score is the single-thread rating[2](#_bookmark27) of each computational configuration. GATS-PR was implemented in

processor whose single-thread rating *𝑠𝑏𝑎𝑠𝑖𝑐* is 2635 MOps/Sec. All CPU C++11, built and executed on an Intel Core i3-10100 @ 3.60 GHz times reported in this section are multiplied by *𝑠𝑎𝑙𝑔* ∕*𝑠𝑏𝑎𝑠𝑖𝑐* , where *𝑠𝑎𝑙𝑔* is the single-thread rating of the CPU used by the algorithm *𝑎𝑙𝑔*. For

GVTS-PR, each instance was independently tested three times with the time limit of one hour.

In terms of *𝑡𝑎𝑣𝑔* , the result of an algorithm is better than/worse

than/equal to that of GATS-PR when:

* + - Its quality is not worse than/not better than/equal to that of GATS- PR. Because HGA, VTNS and MILS were run multiple times, we use

the final value of the parameter and it is indicated in bold in Tables [2](#_bookmark28)

and [3](#_bookmark29).

*𝑓𝑎𝑣𝑔*

as the evaluation criterion of the solution quality. For other

Five parameters are critical to the performance of GATS:

* + - The number *𝛿* of customers considered at each iteration. Its default value is 0*.*3 × |*𝑉 𝐶* |.
    - The combination of the lower and upper bounds *𝑤𝑚𝑖𝑛* and *𝑤𝑚𝑎𝑥* of

neighborhood weights. Its default value is (1, 100).

* + - The discount factor Θ of accumulative generalized objective func-

tion value improvement. Its default value is 0.7.

* + - The convergence speed factor *𝜆* of weights. Its default value is 0.05.

thresholds *𝑔𝑡𝑚𝑖𝑛* and *𝑔𝑡𝑚𝑎𝑥* . Its default value is (0.05, 0.2). • The combination of the minimum and maximum granularity

ber *𝑖𝑡𝑒𝑢𝑛𝐼 𝑚𝑝* of the consecutive iterations without improvement reached The termination condition of the GATS procedure is that the num-

100 × |*𝑉 𝐶* |. The results of tuning GATS-related parameters are shown

reference algorithms except for TS, we use *𝑓𝑏𝑒𝑠𝑡* because they were

in the literature, so we do not compare its *𝑡𝑎𝑣𝑔* with ours. run only once. The computational configuration of TS is not given

* + - Its *𝑡𝑎𝑣𝑔* is smaller than/larger than/equal to that of GATS-PR.

In Tables [6](#_bookmark32) and [7](#_bookmark35), results in column BKS with an asterisk indicate that they are the optimal solutions proved by [[11](#_bookmark64)] and [[12](#_bookmark66)]. Results

tained by GATS-PR. If *𝑓𝑏𝑒𝑠𝑡* or *𝑓𝑎𝑣𝑔* obtained by an algorithm is equal to with an underline indicate that they are new best known results ob- BKS, it is indicated in bold. If *𝑓𝑏𝑒𝑠𝑡* , *𝑓𝑎𝑣𝑔* or *𝑡𝑎𝑣𝑔* obtained by an algorithm

is better than that of GATS-PR, it is shown in italics. Rows *Better*, *Equal*

and *Worse* represent the number of results whose *𝑓𝑏𝑒𝑠𝑡* , *𝑓𝑎𝑣𝑔* or *𝑡𝑎𝑣𝑔* is

better than, equal to or worse than that of GATS-PR, respectively.

Table [6](#_bookmark32) presents *𝑓𝑏𝑒𝑠𝑡* and *𝑡𝑎𝑣𝑔* of GATS-PR and the state-of-the-art

exact algorithms. Except for controversial results produced by IMIP on

instance p15 and Model on instance pr02, *𝑓* obtained by GATS-PR

in Table [2](#_bookmark28). According to *𝑆𝐺𝑏𝑒𝑠𝑡*

, we set *𝛿* to 0*.*5 × |*𝑉 𝐶* |, *𝑤*

*𝑚𝑖𝑛*

to 1, *𝑤*

*𝑚𝑎𝑥*

*𝑏𝑒𝑠𝑡*

to 50, Θ to 0.9, *𝜆* to 0.1, *𝑔𝑡𝑚𝑖𝑛* to 0.05 and *𝑔𝑡𝑚𝑎𝑥* to 0.2.

There are four key parameters for the path relinking operator: 2 <https://www.passmark.com/>.

**Table 2**

The results of the selected instances to tune GATS-related parameters.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| parameter | value | p09 |  |  |  | p10 |  |  |  | pr05 |  |  | *𝑆𝐺𝑏𝑒𝑠𝑡* |
|  |  | *𝑓𝑏𝑒𝑠𝑡* | *𝑓𝑎𝑣𝑔* | *𝑡𝑎𝑣𝑔* |  | *𝑓𝑏𝑒𝑠𝑡* | *𝑓𝑎𝑣𝑔* | *𝑡𝑎𝑣𝑔* |  | *𝑓𝑏𝑒𝑠𝑡* | *𝑓𝑎𝑣𝑔* | *𝑡𝑎𝑣𝑔* |  |
|  | 0*.*1 × |*𝑉 𝐶* | | 2579.74 | 2591.39 | 55.33 |  | 2485.79 | 2490.23 | 40.00 |  | 1704.66 | 1706.62 | 14.00 | 5.81 |
|  | 0*.*3 × |*𝑉 𝐶* | | 2587.74 | 2591.75 | 427.00 |  | 2484.93 | 2487.04 | 157.33 |  | 1695.04 | 1698.22 | 114.00 | 2.90 |
| *𝛿* | **𝟎*.*𝟓× |*𝑽 𝑪* |** | **2577.27** | **2582.25** | **827.67** |  | **2485.09** | **2489.36** | **501.67** |  | **1692.98** | **1696.92** | **241.67** | **-2.31** |
|  | 0*.*7 × |*𝑉 𝐶* | | 2585.91 | 2589.18 | 606.00 |  | 2481.73 | 2488.88 | 726.67 |  | 1700.20 | 1702.05 | 169.33 | 3.94 |
|  | |*𝑉 𝐶* | | 2587.62 | 2591.90 | 747.67 |  | 2491.18 | 2495.77 | 489.33 |  | 1692.73 | 1698.32 | 387.33 | 4.01 |
|  | **(1, 50)** | **2574.00** | **2585.48** | **228.67** |  | **2486.33** | **2492.42** | **118.33** |  | **1692.22** | **1698.91** | **88.67** | **-3.52** |
|  | (1, 100) | 2584.44 | 2589.45 | 306.33 |  | 2485.77 | 2488.66 | 82.67 |  | 1694.62 | 1697.85 | 130.67 | 1.71 |
| (*𝑤𝑚𝑖𝑛* , *𝑤𝑚𝑎𝑥* ) | (1, 500) | 2589.24 | 2593.68 | 193.00 |  | 2480.30 | 2483.79 | 161.00 |  | 1693.54 | 1699.36 | 52.33 | 0.73 |
|  | (1, 1000) | 2579.98 | 2587.00 | 309.00 |  | 2484.05 | 2485.55 | 92.67 |  | 1696.58 | 1699.78 | 89.67 | 0.44 |
|  | (1, 2000) | 2572.92 | 2579.93 | 174.00 |  | 2485.59 | 2486.36 | 130.33 |  | 1697.49 | 1700.24 | 94.00 | -1.14 |
|  | 0.1 | 2581.14 | 2584.84 | 282.00 |  | 2481.59 | 2485.83 | 92.67 |  | 1692.57 | 1700.48 | 135.00 | -2.46 |
|  | 0.3 | 2577.54 | 2590.23 | 155.33 |  | 2485.71 | 2486.86 | 260.67 |  | 1700.62 | 1701.36 | 84.00 | 2.55 |
| Θ | 0.5 | 2585.67 | 2594.99 | 205.33 |  | 2480.07 | 2484.65 | 175.33 |  | 1699.64 | 1702.73 | 82.33 | 2.85 |
|  | 0.7 | 2583.40 | 2593.53 | 211.33 |  | 2485.71 | 2486.42 | 209.67 |  | 1699.35 | 1701.62 | 92.33 | 4.07 |
|  | **0.9** | **2579.22** | **2585.14** | **228.00** |  | **2478.68** | **2482.94** | **235.00** |  | **1694.97** | **1698.00** | **93.67** | **-2.96** |
|  | 0.01 | 2585.30 | 2591.45 | 214.00 |  | 2482.98 | 2494.14 | 161.67 |  | 1696.39 | 1698.22 | 131.33 | 1.96 |
|  | 0.03 | 2583.66 | 2591.28 | 116.00 |  | 2475.63 | 2488.02 | 172.67 |  | 1702.56 | 1703.97 | 75.00 | 2.00 |
| *𝜆* | 0.05 | 2593.09 | 2598.03 | 191.00 |  | 2491.08 | 2491.67 | 228.33 |  | 1692.98 | 1697.63 | 94.00 | 6.24 |
|  | 0.07 | 2588.27 | 2592.70 | 336.00 |  | 2484.60 | 2487.13 | 153.00 |  | 1695.49 | 1698.13 | 72.00 | 3.24 |
|  | **0.1** | **2575.44** | **2585.95** | **217.67** |  | **2476.81** | **2484.31** | **275.67** |  | **1698.63** | **1702.92** | **91.67** | **-3.03** |
|  | (0.05, 0.1) | 2579.22 | 2580.79 | 92.00 |  | 2491.76 | 2497.15 | 74.00 |  | 1693.80 | 1698.65 | 71.67 | 1.62 |
|  | **(0.05, 0.2)** | **2573.64** | **2577.69** | **232.33** |  | **2485.40** | **2486.73** | **84.00** |  | **1696.05** | **1701.09** | **110.67** | **-1.78** |
| (*𝑔𝑡𝑚𝑖𝑛* , *𝑔𝑡𝑚𝑎𝑥* ) | (0.05, 0.4) | 2587.21 | 2590.70 | 266.33 |  | 2479.50 | 2486.30 | 188.33 |  | 1694.70 | 1698.28 | 102.33 | 0.31 |
|  | (0.1, 0.2) | 2578.74 | 2593.27 | 142.67 |  | 2485.81 | 2491.79 | 94.67 |  | 1699.12 | 1700.07 | 108.00 | 2.17 |
|  | (0.1, 0.4) | 2579.88 | 2583.94 | 388.00 |  | 2481.96 | 2483.76 | 141.33 |  | 1695.04 | 1697.58 | 151.33 | -1.34 |

**Table 3**

The results of the selected instances to tune path relinking-related parameters.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| parameter | value | p09 |  |  |  | p10 |  |  |  | pr05 |  |  | *𝑆𝐺𝑏𝑒𝑠𝑡* |
|  |  | *𝑓𝑏𝑒𝑠𝑡* | *𝑓𝑎𝑣𝑔* | *𝑡𝑎𝑣𝑔* |  | *𝑓𝑏𝑒𝑠𝑡* | *𝑓𝑎𝑣𝑔* | *𝑡𝑎𝑣𝑔* |  | *𝑓𝑏𝑒𝑠𝑡* | *𝑓𝑎𝑣𝑔* | *𝑡𝑎𝑣𝑔* |  |
|  | 1 | 2577.71 | 2580.60 | 550.67 |  | 2482.98 | 2483.93 | 492.00 |  | 1695.20 | 1696.65 | 349.00 | -1.68 |
|  | 2 | 2582.50 | 2585.38 | 616.33 |  | 2476.12 | 2481.61 | 418.67 |  | 1700.72 | 1701.92 | 389.33 | 0.67 |
| *𝛾* | 3 | 2577.40 | 2579.83 | 611.67 |  | 2478.86 | 2482.98 | 553.67 |  | 1695.04 | 1698.00 | 390.33 | -3.55 |
|  | **5** | **2580.61** | **2584.18** | **560.67** |  | **2474.82** | **2476.47** | **432.33** |  | **1693.52** | **1695.42** | **552.33** | **-4.83** |
|  | 10 | 2578.84 | 2583.65 | 583.00 |  | 2476.85 | 2479.00 | 414.33 |  | 1696.54 | 1699.14 | 517.00 | -2.92 |
|  | 0.1 | 2579.20 | 2585.21 | 569.33 |  | 2474.82 | 2481.41 | 519.00 |  | 1695.04 | 1696.32 | 401.33 | -4.48 |
|  | 0.3 | 2575.36 | 2577.63 | 537.67 |  | 2477.38 | 2482.45 | 558.67 |  | 1694.97 | 1697.53 | 456.00 | -4.98 |
| *𝑤𝑆𝐷* | 0.5 | 2578.41 | 2584.23 | 517.67 |  | 2476.49 | 2480.26 | 484.33 |  | 1696.26 | 1696.30 | 454.33 | -3.40 |
|  | **0.7** | **2575.99** | **2583.97** | **555.67** |  | **2474.82** | **2479.97** | **482.33** |  | **1694.62** | **1697.48** | **538.00** | **-5.98** |
|  | 0.9 | 2578.38 | 2591.81 | 615.33 |  | 2476.49 | 2479.31 | 494.33 |  | 1697.68 | 1699.41 | 618.33 | -2.57 |
|  | **(10, 1)** | **2577.20** | **2578.70** | **459.67** |  | **2476.12** | **2477.29** | **430.33** |  | **1692.22** | **1696.63** | **300.00** | **-6.40** |
|  | (10, 2) | 2585.22 | 2589.46 | 519.67 |  | 2478.85 | 2482.84 | 578.33 |  | 1693.26 | 1698.00 | 557.33 | -1.57 |
| (*𝜇*, *𝑖𝑡𝑒𝑚𝑎𝑥* ) | (20, 1) | 2578.01 | 2580.88 | 692.67 |  | 2476.12 | 2480.06 | 635.33 |  | 1693.24 | 1695.31 | 493.33 | -5.48 |
|  | (20, 2) | 2580.01 | 2586.31 | 559.00 |  | 2483.26 | 2483.89 | 620.67 |  | 1696.13 | 1697.34 | 537.00 | -0.13 |
|  | (40, 1) | 2579.17 | 2584.75 | 787.33 |  | 2476.12 | 2477.76 | 843.67 |  | 1696.55 | 1698.78 | 717.00 | -3.08 |
|  | (0.7, 0.3) | 2580.23 | 2585.00 | 605.67 |  | 2476.12 | 2478.92 | 599.33 |  | 1694.30 | 1698.29 | 414.00 | -4.00 |
|  | (0.6, 0.4) | 2577.40 | 2582.39 | 549.33 |  | 2479.24 | 2484.17 | 471.67 |  | 1696.39 | 1698.49 | 422.33 | -2.61 |
| (*𝑟𝑎𝑡𝑖𝑜𝑏𝑒𝑔* , *𝑟𝑎𝑡𝑖𝑜𝑒𝑛𝑑* ) | (0.6, 0.3) | 2582.37 | 2588.81 | 553.33 |  | 2475.20 | 2479.26 | 588.00 |  | 1697.92 | 1701.39 | 513.33 | -1.40 |
|  | (0.6, 0.2) | 2577.22 | 2584.49 | 499.33 |  | 2475.20 | 2479.40 | 442.00 |  | 1696.11 | 1699.52 | 287.00 | -4.47 |
|  | **(0.5, 0.2)** | **2578.48** | **2582.43** | **601.00** |  | **2475.20** | **2480.04** | **624.67** |  | **1691.94** | **1695.51** | **385.33** | **-6.44** |

is not worse than that of the state-of-the-art exact algorithms on all the instances. Specifically, GATS-PR improved BKS on four instances

p09, p10, p11 and pr05. In terms of average *𝐺𝑎𝑝𝑏𝑒𝑠𝑡* and *𝑡𝑎𝑣𝑔* , GATS-PR

obtains significantly better results than the reference algorithms in less

than 30% run time of Model, which is the most eﬃcient exact algorithm among the reference algorithms. Hence, GATS-PR outperforms these state-of-the-art exact algorithms.

Tables [7](#_bookmark35) and [8](#_bookmark40) report *𝑓𝑏𝑒𝑠𝑡* , *𝑓𝑎𝑣𝑔* and *𝑡𝑎𝑣𝑔* of GATS-PR and the state-

of-the-art metaheuristic algorithms.

* + - In terms of *𝑓𝑏𝑒𝑠𝑡* , MNS-TS obtains 6 worse solutions than GATS-PR and its average *𝐺𝑎𝑝𝑏𝑒𝑠𝑡* (i.e. 0.43) is worse than that of GATS-PR

on instances p12, p15 and p18, GATS-PR obtains better *𝑓𝑏𝑒𝑠𝑡* than (i.e. -0.60). Except for controversial results produced by MNS-TS

**Table 4**

Experimental environment of the state-of-the-art exact algo- rithms.

|  |  |  |
| --- | --- | --- |
| Name | CPU | Score (MOps/Sec) |
| IMIP [[16](#_bookmark73)] | Intel Dual Core @ 3.50 GHz | 1954 |
| TIVF [[10](#_bookmark62)] | Intel Dual Core @ 3.50 GHz | 1954 |
| TIMIP [[11](#_bookmark64)] | Intel Core i7 @ 3.70 GHz | 2759 |
| Model [[12](#_bookmark66)] | Intel Core i5-6300 @ 2.40 GHz | 1676 |

**Table 5**

Experimental environment of the state-of-the-art heuristic algorithms.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Num | CPU | Score (MOps/Sec) |
| HGA [[8](#_bookmark58)] | 20 | Intel Dual Core @ 3.2 GHz | 550 |
| MNS-TS [[26](#_bookmark56)] | 1 | Intel Core i7-2600 @ 3.40 GHz | 1742 |
| MBILSA [[27](#_bookmark59)] | 1 | Intel Core i7-3820 @ 3.60 GHz | 1739 |
| VTNS [[28](#_bookmark61)] | 25 | Intel Core i7-8700 @ 3.20 GHz | 2657 |
| MILS [[12](#_bookmark66)] | 15 | Intel Core i5-6300 @ 2.40 GHz | 1676 |
| TS [[29](#_bookmark63)] | 1 | - | - |

**Table 6**

The comparison of the solution quality of GATS-PR and the state-of-the-art exact algorithms.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ins | BKS | Ref | *𝑓𝑏𝑒𝑠𝑡* |  |  |  |  |  | *𝑡𝑎𝑣𝑔* |  | | | | |
|  |  |  | IMIP | TIVF | TIMIP | Model | GATS-PR |  | IMIP | TIVF | TIMIP | Model | GATS-PR |  |
| p01 | 386.18\* | HGA | **386.18** | **386.18** | **386.18** | **386.18** | **386.18** |  | 50.2 | 20.91 | 0.95 | 2.54 | 0.00 |  |
| p02 | 375.93\* | HGA | **375.93** | **375.93** | **375.93** | **375.93** | **375.93** |  | 0.74 | 0.44 | 0.21 | 0.64 | 0.00 |  |
| p03 | 474.57\* | HGA | **474.57** | **474.57** | **474.57** | **474.57** | **474.57** |  | 50.2 | 13.53 | 1.44 | 3.82 | 1.00 |  |
| p04 | 662.22 | IMIP | **662.22** | **662.22** | **662.22** | **662.22** | **662.22** |  | 5339.2 | 5339.2 | 7538.82 | 4579.58 | 4.00 |  |
| p05 | 607.53\* | TIVF | 608.73 | **607.53** | **607.53** | **607.53** | **607.53** |  | 5339.2 | 33.81 | 76.38 | 22.9 | 8.00 |  |
| p06 | 611.99\* | TIVR | **611.99** | **611.99** | **611.99** | **611.99** | **611.99** |  | 5339.2 | 519.09 | 431.35 | 321.21 | 12.00 |  |
| p07 | 608.28\* | TIVF | 613.96 | **608.28** | **608.28** | **608.28** | **608.28** |  | 5339.2 | 509.45 | 1079.96 | 494.21 | 52.67 |  |
| p08 | 2776.12 | VTNS | 3156.89 | 2898.51 | 2870.21 | 2853.57 | 2776.35 |  | 5339.2 | 5339.2 | 7538.82 | 4579.58 | 2473.00 |  |
| p09 | 2578.49 | MBILSA | 2795.43 | 2627.28 | 2660.45 | 2595.80 | 2572.46 |  | 5339.2 | 5339.2 | 7538.82 | 4579.58 | 1230.33 |  |
| p10 | 2482.32 | VTNS | 2704.64 | 2505.37 | 2528.98 | 2488.77 | 2474.82 |  | 5339.2 | 5339.2 | 7538.82 | 4579.58 | 2798.00 |  |
| p11 | 2465.28 | Model | 2844.66 | 2516.60 | 2499.25 | **2465.28** | 2451.33 |  | 5339.2 | 5339.2 | 7538.82 | 4579.58 | 1108.33 |  |
| p12 | 953.26\* | HGA | **953.26** | **953.26** | **953.26** | **953.26** | **953.26** |  | 0.54 | 0.56 | 0.48 | 1.27 | 0.00 |  |
| p15 | 1885.81\* | TIVF | *1883.53*[1](#_bookmark33) | **1885.81** | **1885.81** | **1885.81** | **1885.81** |  | 31.67 | 13.39 | 3.8 | 11.45 | 0.33 |  |
| p18 | 2818.36\* | IMIP | **2818.36** | **2818.36** | **2818.36** | **2818.36** | **2818.36** |  | 278.75 | 13.73 | 15.37 | 41.34 | 2.33 |  |
| pr01 | 647.03\* | HGA | **647.03** | **647.03** | **647.03** | **647.03** | **647.03** |  | 0.76 | 0.37 | 0.15 | 0.64 | 0.00 |  |
| pr02 | 979.82\* | IMIP | **979.82** | **979.82** | **979.82** | *978.82*[2](#_bookmark34) | **979.82** |  | 16.26 | 4.23 | 2.06 | 3.82 | 0.67 |  |
| pr03 | 1423.48\* | IMIP | **1423.48** | **1423.48** | **1423.48** | **1423.48** | **1423.48** |  | 5339.2 | 28.7 | *17.63* | 23.53 | 21.67 |  |
| pr04 | 1514.07\* | TIMIP | 1522.52 | 1521.69 | **1514.07** | **1514.07** | **1514.07** |  | 5339.2 | 5339.2 | 1819.98 | 866.3 | 28.00 |  |
| pr05 | 1697.99 | Model | 1802.71 | 1711.68 | 1716.02 | **1697.99** | 1691.94 |  | 5339.2 | 5339.2 | 7538.82 | 4579.58 | 843.67 |  |
| pr06 | 1976.47 | Model | 2221.94 | 2014.69 | 1978.46 | **1976.47** | **1976.47** |  | 5339.2 | 5339.2 | 7538.82 | 4579.58 | 2125.00 |  |
| pr07 | 821.25\* | HGA | **821.25** | **821.25** | **821.25** | **821.25** | **821.25** |  | 2.8 | 1.89 | 0.31 | 0.64 | 0.00 |  |
| pr08 | 1254.45\* | IMIP | **1254.45** | **1254.45** | **1254.45** | **1254.45** | **1254.45** |  | 5339.2 | 37.15 | 8.85 | 20.99 | 6.67 |  |
| pr09 | 1591.78\* | TIVF | 1659.39 | **1591.78** | **1591.78** | **1591.78** | **1591.78** |  | 5339.2 | 5339.2 | 341.57 | 526.65 | 64.33 |  |
| pr10 | 1968.67 | Model | 2295.01 | 2051.59 | 1997.96 | **1968.67** | **1968.67** |  | 5339.2 | 5339.2 | 7538.82 | 4579.58 | 579.00 |  |
| *Better* |  |  | 1 | 0 | 0 | 1 | 0 |  | 0 | 0 | 1 | 0 | 0 |  |
| *Equal* |  |  | 12 | 16 | 17 | 20 | 24 |  | 0 | 0 | 0 | 0 | 24 |  |
| *Worse* |  |  | 11 | 8 | 7 | 3 | 0 |  | 24 | 24 | 23 | 24 | 0 |  |
| *𝐴𝑣𝑒* |  |  | 36.44 | 6.99 | 5.20 | 1.51 | -0.60 |  | 3355.00 | 2274.55 | 2671.29 | 1624.11 | 473.29 |  |

1 The same author reported that the objective function value of the optimal solution on p15 is 1885.81 in the subsequent paper [[11](#_bookmark64)].

2 The objective function value of the optimal solution on pr02 is proved to be 979.82 in [[11](#_bookmark64)] and [[16](#_bookmark73)].

HGA, MNS-TS, MBILSA, MILS and TS on all the instances and the values of *Worse* exceed 15 for these algorithms. It indicates that GATS-PR obtains more high-quality solutions than the state-of-the- art metaheuristic algorithms.

* + - In terms of *𝑓𝑎𝑣𝑔* , GATS-PR obtains significantly better results than

HGA and VTNS in terms of average *𝐺𝑎𝑝𝑎𝑣𝑔* . Specifically, not only

our *𝑓𝑎𝑣𝑔* on these instances was better than BKS. Although VTNS we improved BKS on instances p09, p10, p11 and pr05, but also

terms of *𝑓𝑏𝑒𝑠𝑡* , its *𝑓𝑎𝑣𝑔* (i.e. 3036.02) is significantly worse than that obtains a slightly better solution than GATS-PR on instance p08 in

of GATS-PR (i.e. 2778.26). It indicates that GATS-PR can obtain high-quality solutions frequently and GATS-PR is quite robust.

* + - In terms of *𝑡𝑎𝑣𝑔* , no reference algorithms can obtain better results in

less time than GATS-PR. The values of *Worse* are 16, 8, 15, 16 and 13 for HGA, MNS-TS, MBILSA, VTNS and MILS. It shows that GATS- PR is more eﬃcient than them. Although GATS-PR spends a lot of time on large-scale instances, such as p08 and p10, it indicates that GATS-PR has better search capability than the reference algorithms because GATS-PR can obtain better results than them.

**Table 7**

The comparison of the results of GATS-PR and the state-of-the-art metaheuristic algorithms.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ins | BKS | Ref | *𝑓𝑏𝑒𝑠𝑡* |  |  |  |  |  |  |  | *𝑓𝑎𝑣𝑔* |  | | |
|  |  |  | HGA | MNS-TS | MBILSA | VTNS | MILS[3](#_bookmark38) | TS | GATS-PR |  | HGA | VTNS | GATS-PR |  |
| p01 | 386.18\* | HGA | **386.18** | **386.18** | **386.18** | **386.18** | **386.18** | 386.69 | **386.18** |  | 388.82 | 386.95 | **386.18** |  |
| p02 | 375.93\* | HGA | **375.93** | 376.78 | 376.44 | **375.93** | **375.93** | 377.76 | **375.93** |  | 376.90 | 376.04 | **375.93** |  |
| p03 | 474.57\* | HGA | **474.57** | **474.57** | 475.79 | **474.57** | 485.20 | 481.56 | **474.57** |  | 477.69 | 474.73 | **474.57** |  |
| p04 | 662.22 | IMIP | 667.02 | 670.13 | **662.22** | **662.22** | 689.14 | 686.55 | **662.22** |  | 669.49 | 663.83 | **662.22** |  |
| p05 | 607.53\* | TIVF | 612.30 | 609.81 | 609.04 | **607.53** | 623.03 | 617.87 | **607.53** |  | 616.35 | 608.88 | **607.53** |  |
| p06 | 611.99\* | TIVR | 614.93 | 616.90 | **611.99** | **611.99** | 628.72 | 631.69 | **611.99** |  | 621.13 | 614.60 | **611.99** |  |
| p07 | 608.28\* | TIVF | 615.24 | 615.98 | 609.60 | **608.28** | 631.60 | 626.11 | **608.28** |  | 623.91 | 610.82 | **608.28** |  |
| p08 | 2776.12 | VTNS | 2,913.63[1](#_bookmark36) | 2852.56 | 2794.10 | ***2776.12*** | 3027.15 | - | 2776.35 |  | 2,911.51[1](#_bookmark36) | 3036.02 | 2778.26 |  |
| p09 | 2578.49 | MBILSA | 2,738.54 | 2625.04 | 2582.65 | 2587.04 | 3049.10 | - | 2572.46 |  | 2,755.28 | 2619.16 | 2573.43 |  |
| p10 | 2482.32 | VTNS | 2,577.48 | 2539.60 | 2500.43 | **2482.32** | 2730.14 | - | 2474.82 |  | 2,616.02 | 2500.41 | 2475.25 |  |
| p11 | 2465.28 | Model | 2,561.85 | 2508.45 | 2478.31 | 2472.85 | 2707.01 | - | 2451.33 |  | 2,598.56 | 2487.38 | 2456.92 |  |
| p12 | 953.26\* | HGA | **953.26** | *953.25*[2](#_bookmark37) | **953.26** | **953.26** | 1021.16 | **953.26** | **953.26** |  | 955.80 | **953.26** | **953.26** |  |
| p15 | 1885.81\* | TIVF | 1,888.67 | *1885.80*[2](#_bookmark37) | **1885.81** | **1885.81** | 2044.42 | 1932.51 | **1885.81** |  | 1,899.98 | **1885.81** | **1885.81** |  |
| p18 | 2818.36\* | IMIP | 2,830.78 | *2818.34*[2](#_bookmark37) | **2818.36** | **2818.36** | 2939.89 | - | **2818.36** |  | 2,854.81 | **2818.36** | **2818.36** |  |
| pr01 | 647.03\* | HGA | **647.03** | **647.03** | **647.03** | **647.03** | 768.43 | **647.03** | **647.03** |  | **647.03** | **647.03** | **647.03** |  |
| pr02 | 979.82\* | IMIP | 981.63 | **979.82** | **979.82** | **979.82** | 992.41 | 1004.84 | **979.82** |  | 985.69 | **979.82** | **979.82** |  |
| pr03 | 1423.48\* | IMIP | 1,445.65 | 1,426.95 | 1429.38 | **1423.48** | 1475.74 | 1451.17 | **1423.48** |  | 1,458.75 | 1426.68 | **1423.48** |  |
| pr04 | 1514.07\* | TIMIP | 1,548.93 | 1,517.80 | 1524.18 | **1514.07** | 1637.30 | - | **1514.07** |  | 1,586.67 | 1519.49 | **1514.07** |  |
| pr05 | 1697.99 | Model | 1,746.57 | 1,758.51 | 1700.93 | 1699.40 | 1915.10 | - | 1691.94 |  | 1,755.26 | 1705.98 | 1691.94 |  |
| pr06 | 1976.47 | Model | 2,069.01 | 2009.97 | 1992.26 | 1982.13 | 2152.19 | - | **1976.47** |  | 2,085.62 | 1992.32 | 1976.98 |  |
| pr07 | 821.25\* | HGA | **821.25** | **821.25** | **821.25** | **821.25** | 839.35 | 830.78 | **821.25** |  | 824.09 | **821.25** | **821.25** |  |
| pr08 | 1254.45\* | IMIP | 1,266.87 | 1,266.17 | 1258.64 | **1254.45** | 1325.12 | 1283.43 | **1254.45** |  | 1277.79 | 1255.40 | **1254.45** |  |
| pr09 | 1591.78\* | TIVF | 1,643.56 | 1,615.42 | 1592.77 | **1591.78** | 1694.34 | - | **1591.78** |  | 1666.02 | 1592.83 | **1591.78** |  |
| pr10 | 1968.67 | Model | 2,037.22 | 2013.86 | 1975.36 | 1969.35 | 2186.16 | - | **1968.67** |  | 2076.08 | 1979.00 | **1968.67** |  |
| *Better* |  |  | 0 | 3 | 0 | 1 | 0 | 0 | 0 |  | 0 | 0 | 0 |  |
| *Equal* |  |  | 6 | 5 | 9 | 17 | 2 | 2 | 24 |  | 1 | 6 | 24 |  |
| *Worse* |  |  | 18 | 16 | 15 | 6 | 22 | 22 | 0 |  | 23 | 18 | 0 |  |
| *𝐴𝑣𝑒* |  |  | 17.48 | 9.56 | 2.38 | 0.43 | 67.17 | -[4](#_bookmark39) | -0.60 |  | 25.89 | 6.94 | -0.45 |  |

1 These two values are probably recorded in reverse in [[8](#_bookmark58)].

2 The results are better than the optimal ones proved by [[10](#_bookmark62)] and [[11](#_bookmark64)].

3 We round the results in [[12](#_bookmark66)] because results in other literature are only retained to two decimal places.

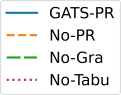
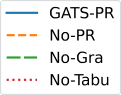
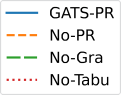
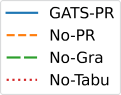
4 The value is missing because TS cannot obtain feasible solutions on a lot of instances.

**Table 8**

The comparison of *𝑡𝑎𝑣𝑔* of GATS-PR and the state-of-the-art metaheuristic

algorithms.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Ins |  | HGA | MNS-TS | MBILSA | VTNS | MILS | GATS-PR |  |
| p01 |  | 2.88 | 0.97 | 5.02 | 16.44 | 2.89 | 0.00 |  |
| p02 |  | 3.03 | 0.81 | 5.74 | 21.77 | 1.66 | 0.00 |  |
| p03 |  | 8.52 | 3.99 | 10.69 | 40.14 | 4.85 | 1.00 |  |
| p04 |  | 17.95 | 3.97 | 20.19 | 15.88 | 67.24 | 4.00 |  |
| p05 |  | 18.99 | 9.80 | 14.65 | 31.13 | 7.60 | 8.00 |  |
| p06 |  | 19.08 | 17.35 | 14.52 | 23.51 | 13.10 | 12.00 |  |
| p07 |  | 18.31 | 14.20 | 20.19 | 28.89 | 29.89 | 52.67 |  |
| p08 |  | 227.39 | 36.62 | 97.74 | 18.11 | 78.77 | 2473.00 |  |
| p09 |  | 230.08 | 39.54 | 100.05 | 99.08 | 85.86 | 1230.33 |  |
| p10 |  | 233.53 | 32.14 | 76.09 | 188.99 | 83.50 | 2798.00 |  |
| p11 |  | 232.48 | 56.22 | 87.97 | 252.65 | 85.51 | 1108.33 |  |
| p12 |  | 10.29 | 0.80 | 6.86 | 42.40 | 4.16 | 0.00 |  |
| p15 |  | 70.34 | 6.07 | 24.15 | 159.13 | 25.37 | 0.33 |  |
| p18 |  | 216.7 | 4.40 | 50.16 | 360.23 | 67.42 | 2.33 |  |
| pr01 |  | 2.73 | 0.37 | 3.30 | 42.84 | 1.29 | 0.00 |  |
| pr02 |  | 17.72 | 6.00 | 12.61 | 85.69 | 6.34 | 0.67 |  |
| pr03 |  | 63.10 | 18.74 | 27.85 | 152.21 | 17.72 | 21.67 |  |
| pr04 |  | 76.81 | 9.31 | 39.93 | 208.32 | 36.40 | 28.00 |  |
| pr05 |  | 213.82 | 63.43 | 85.73 | 281.21 | 64.06 | 843.67 |  |
| pr06 |  | 368.93 | 45.01 | 118.33 | 311.95 | 110.47 | 2125.00 |  |
| pr07 |  | 8.50 | 0.78 | 7.79 | 97.58 | 3.43 | 0.00 |  |
| pr08 |  | 52.54 | 3.72 | 21.38 | 188.57 | 17.74 | 6.67 |  |
| pr09 |  | 162.50 | 11.47 | 56.56 | 309.50 | 51.32 | 64.33 |  |
| pr10 |  | 363.38 | 116.00 | 115.16 | 446.58 | 112.33 | 579.80 |  |
| *Better* |  | 0 | 0 | 0 | 0 | 0 | 0 |  |
| *Equal* |  | 0 | 0 | 0 | 0 | 0 | 0 |  |
|  | *Worse* |  | 16 | 8 | 15 | 16 | 13 | 0 |  |



**Fig. 4.** Evolution process of *𝑓𝑏𝑒𝑠𝑡* for GATS-PR, No-PR, No-Gra and No-Tabu over time on a logarithmic scale.

These comparisons demonstrate that GATS-PR outperforms these state- of-the-art metaheuristic algorithms for solving the MDOVRP.

* 1. *Importance analysis of algorithmic components*

In order to analyze the importance of the proposed algorithmic com- ponents, we disable one of them from GATS-PR at a time and compare the results obtained by the variant algorithm with those of GATS-PR. The No-PR version only uses the granular adaptive tabu search pre- sented in Algorithm [2](#_bookmark11) to verify the effectiveness of path relinking. The

number of neighboring nodes of a given customer by setting both *𝑔𝑡𝑚𝑖𝑛* No-Gra version disables granular neighborhoods, which considers fixed and *𝑔𝑡𝑚𝑎𝑥* to 0.2. The No-Tabu version chooses the move with the best

improvement of the generalized function value at each iteration to val- idate the role of the solution-based tabu search. The No-ANS version does not use the adaptive neighborhood selection (ANS) but considers all the move types described in Fig. [2](#_bookmark7) at each iteration. The No-Group version regards each move type as a neighborhood to verify the effec- tiveness of grouping move types into four neighborhoods in ANS.

For GATS-PR and its variants, each instance was independently run three times and the time limit was one hour. The results of these algo- rithms are shown in Table [9](#_bookmark42). Although No-Tabu converges quickly, its results are significantly worse than those of GATS-PR. It indicates that the solution-based tabu search enables the search to escape from the local optima. GATS-PR obtains better results with less run time than No-PR and No-Gra. It demonstrates that path relinking and granular neighborhoods can improve the search eﬃciency without sacrificing the solution quality. No-Group obtains better solutions than No-ANS with less run time and GATS-PR obtains better solutions than No-Group with less time. It demonstrates that ANS can improve eﬃciency and ef- fectiveness of the search and grouping multiple move types into one neighborhood contributes to ANS when the number of move types is large. Therefore, all these algorithmic components are important for GATS-PR.

variants, Figs. [4](#_bookmark41) and [5](#_bookmark43) plot the evolution process of *𝑓𝑏𝑒𝑠𝑡* obtained by In order to intuitively show the convergence of GATS-PR and its

each algorithm on four representative instances p08, p09, pr05 and pr10 over time on a logarithmic scale. As shown in Figs. [4](#_bookmark41) and [5](#_bookmark43), we observe that:

* Although GATS-PR obtains slightly worse solutions than No-PR and No-Tabu at the early stage of the search, GATS-PR outperforms them on instances p08 and p09, and match them on instances pr05 and pr10 at the end of the search. It indicates that GATS-PR con- verges slightly slower than No-PR and No-Tabu but it has a better search capability.
* After excluding solutions obtained at the earliest stage of the search which are heavily influenced by initial solutions and with cer- tain randomness, GATS-PR outperforms No-Gra on the selected instances. It shows that the granular neighborhoods can improve the eﬃciency of the search without sacrificing the solution quality.
* After excluding solutions obtained at the earliest stage of the search, GATS-PR always outperforms No-ANS on these four in- stances. Although No-Group obtains better solutions than GATS- PR at the middle stage of the search on instance pr05, it obtains worse solutions at the end of the search. On other three instances, No-Group is outperformed by GATS-PR. These observations demon- strate the importance of ANS again.

To further explore the effectiveness of ANS on the neighborhood se- lection mechanism, we record the frequency of performing each move type when the best solution is obtained by No-ANS, No-Group and

GATS-PR, respectively. Let *𝑟𝑖, 𝑖* ∈ *𝑁* represent the ratio of the number

of chosen move types that belong to neighborhood *𝑖* to the total num- ber of iterations. Fig. [6](#_bookmark45) presents *𝑟𝑖* of these three algorithms on all the

instances.

*W. Shao, T. Xiao, Z. Su et al.*

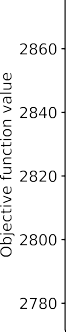
*Egyptian Informatics Journal 24 (2023) 100420*

11

**Table 9**

Results of the GATS-PR and its variants.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Ins |  | *𝑓𝑏𝑒𝑠𝑡* |  |  |  |  |  |  | *𝑓𝑎𝑣𝑔* |  |  |  |  |  |  | *𝑡𝑎𝑣𝑔* |  | | | | | |
|  |  | GATS-PR | No-PR | No-Gra | No-Tabu | No-ANS | No-Group |  | GATS-PR | No-PR | No-Gra | No-Tabu | No-ANS | No-Group |  | GATS-PR | No-PR | No-Gra | No-Tabu | No-ANS | No-Group |  |
| p01 |  | 386.18 | 386.18 | 386.18 | 386.18 | 386.18 | 386.18 |  | 386.18 | 386.18 | 386.18 | 386.18 | 386.18 | 386.18 |  | 0.00 | 2.00 | 4.33 | 0.00 | 7.33 | 7.33 |  |
| p02 |  | 375.93 | 375.93 | 375.93 | 375.93 | 375.93 | 375.93 |  | 375.93 | 375.93 | 375.93 | 375.93 | 375.93 | 375.93 |  | 0.00 | 1.00 | 4.33 | 0.00 | 1.33 | 5.33 |  |
| p03 |  | 474.57 | 474.57 | 474.57 | 474.57 | 474.57 | 474.57 |  | 474.57 | 474.57 | 474.57 | 474.57 | 474.57 | 474.57 |  | 1.00 | 4.00 | 21.33 | 5.67 | 24.67 | 20.67 |  |
| p04 |  | 662.22 | 662.22 | 662.22 | 662.22 | 662.22 | 662.22 |  | 662.22 | 662.22 | 662.43 | 662.22 | 662.22 | 662.22 |  | 4.00 | 8.33 | 45.33 | 18.67 | 41.33 | 183.67 |  |
| p05 |  | 607.53 | 607.53 | 607.53 | 607.53 | 607.53 | 607.53 |  | 607.53 | 607.81 | 607.53 | 607.65 | 608.20 | 609.39 |  | 8.00 | 12.67 | 309.67 | 36.67 | 56.67 | 76.67 |  |
| p06 |  | 611.99 | 611.99 | 611.99 | 611.99 | 611.99 | 611.99 |  | 611.99 | 611.99 | 612.74 | 611.99 | 613.49 | 612.65 |  | 12.00 | 15.33 | 32.67 | 6.33 | 24.33 | 72.00 |  |
| p07 |  | 608.28 | 608.28 | 608.28 | 608.28 | 608.28 | 608.28 |  | 608.28 | 608.28 | 608.95 | 608.28 | 609.25 | 609.13 |  | 52.67 | 18.00 | 74.00 | 27.67 | 51.67 | 60.00 |  |
| p08 |  | 2776.35 | 2782.30 | 2783.11 | 2777.58 | 2791.88 | 2782.31 |  | 2778.26 | 2782.31 | 2793.90 | 2781.18 | 2793.80 | 2784.67 |  | 2473.00 | 2491.00 | 3625.00 | 2122.00 | 2536.00 | 3233.00 |  |
| p09 |  | 2572.46 | 2573.12 | 2575.27 | 2572.46 | 2576.25 | 2577.56 |  | 2573.43 | 2576.21 | 2575.33 | 2573.45 | 2581.16 | 2581.84 |  | 1230.33 | 357.67 | 536.33 | 876.33 | 1340.67 | 2210.33 |  |
| p10 |  | 2474.82 | 2475.49 | 2475.95 | 2474.82 | 2475.49 | 2476.12 |  | 2475.25 | 2476.40 | 2479.54 | 2475.68 | 2476.03 | 2480.04 |  | 2798.00 | 2902.00 | 2997.33 | 1629.33 | 3162.00 | 2415.33 |  |
| p11 |  | 2451.33 | 2451.33 | 2460.50 | 2464.55 | 2461.94 | 2454.45 |  | 2456.92 | *2452.23* | 2462.06 | 2477.88 | 2463.30 | 2457.62 |  | 1108.33 | 2012.67 | 2695.67 | 84.67 | 2869.67 | 1731.00 |  |
| p12 |  | 953.26 | 953.26 | 953.26 | 953.26 | 953.26 | 953.26 |  | 953.26 | 953.26 | 953.26 | 953.26 | 953.26 | 953.26 |  | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.33 |  |
| p15 |  | 1885.81 | 1885.81 | 1885.81 | 1885.81 | 1885.81 | 1885.81 |  | 1885.81 | 1885.81 | 1885.81 | 1885.81 | 1885.81 | 1885.81 |  | 0.33 | 0.33 | 3.67 | 0.67 | 2.00 | 58.33 |  |
| p18 |  | 2818.36 | 2818.36 | 2818.36 | 2818.36 | 2818.36 | 2818.36 |  | 2818.36 | 2818.36 | 2818.36 | 2818.36 | 2818.36 | 2821.12 |  | 2.33 | 1.33 | 14.33 | 1.67 | 6.67 | 44.33 |  |
| pr01 |  | 647.03 | 647.03 | 647.03 | 647.03 | 647.03 | 647.03 |  | 647.03 | 647.03 | 647.03 | 647.03 | 647.03 | 647.03 |  | 0.00 | 0.00 | 1.00 | 0.00 | 1.00 | 0.00 |  |
| pr02 |  | 979.82 | 979.82 | 979.82 | 979.82 | 979.82 | 979.82 |  | 979.82 | 980.36 | 979.82 | 979.82 | 980.34 | 979.82 |  | 0.67 | 6.67 | 23.67 | 8.33 | 14.33 | 29.33 |  |
| pr03 |  | 1423.48 | 1423.48 | 1423.48 | 1423.48 | 1423.48 | 1423.48 |  | 1423.48 | 1423.48 | 1423.48 | 1423.48 | 1423.48 | 1424.79 |  | 21.67 | 43.67 | 189.67 | 31.67 | 224.33 | 77.33 |  |
| pr04 |  | 1514.07 | 1514.07 | 1514.07 | 1514.07 | 1514.07 | 1514.07 |  | 1514.07 | 1514.07 | 1514.07 | 1514.07 | 1514.07 | 1514.07 |  | 28.00 | 2032.33 | 2880.67 | 663.33 | 1712.00 | 1096.33 |  |
| pr05 |  | 1691.94 | 1692.22 | 1691.94 | 1691.94 | 1693.50 | 1692.44 |  | 1691.94 | 1694.68 | 1693.54 | 1693.59 | 1696.13 | 1694.05 |  | 843.67 | 1407.67 | 1750.00 | 2068.33 | 1538.33 | 1466.67 |  |
| pr06 |  | 1976.47 | 1976.47 | 1977.41 | 1976.47 | 1977.41 | 1977.07 |  | 1976.98 | 1977.03 | 1980.62 | 1977.09 | 1978.28 | 1978.54 |  | 2125.00 | 2079.67 | 2955.67 | 793.00 | 3009.33 | 1697.00 |  |
| pr07 |  | 821.25 | 821.25 | 821.25 | 821.25 | 821.25 | 821.25 |  | 821.25 | 821.25 | 821.25 | 821.25 | 821.25 | 821.25 |  | 0.00 | 0.00 | 3.67 | 0.00 | 1.67 | 1.33 |  |
| pr08 |  | 1254.45 | 1254.45 | 1254.45 | 1254.45 | 1254.45 | 1254.45 |  | 1254.45 | 1254.54 | 1255.07 | 1254.45 | 1255.31 | 1254.45 |  | 6.67 | 28.00 | 119.33 | 13.67 | 71.00 | 35.33 |  |
| pr09 |  | 1591.78 | 1591.78 | 1591.78 | 1591.78 | 1591.78 | 1591.78 |  | 1591.78 | 1591.78 | 1591.78 | 1591.78 | 1591.78 | 1591.78 |  | 64.33 | 10.33 | 38.33 | 62.67 | 345.00 | 154.67 |  |
| pr10 |  | 1968.67 | 1968.67 | 1968.67 | 1968.67 | 1969.79 | 1968.67 |  | 1968.67 | 1968.67 | 1968.67 | 1968.67 | 1970.50 | 1968.67 |  | 579.00 | 808.67 | 838.33 | 527.67 | 1136.33 | 223.67 |  |
|  | *𝐴𝑣𝑒* |  | -0.60 | -0.49 | -0.26 | -0.36 | -0.04 | -0.33 |  | -0.45 | -0.29 | 0.22 | 0.01 | 0.47 | 0.27 |  | 473.29 | 593.47 | 798.51 | 374.10 | 757.40 | 620.87 |  |



**Fig. 5.** Evolution process of *𝑓𝑏𝑒𝑠𝑡* for GATS-PR, No-ANS and No-Group over time on a logarithmic scale.

* For No-ANS, *𝑟𝑁*1 exceeds 50% on 19 out of 24 instances and it ex- ceeds 70% on 10 ones. For GATS-PR, although *𝑁*1 is also the most frequently performed neighborhood except for instance p02, *𝑟𝑁*1

is only about 45%. Because GATS-PR obtains better solutions than No-ANS, it indicates that ANS diversifies the search at the suitable

time by selecting neighborhoods *𝑁*3 and *𝑁*4 more frequently than

No-ANS to help the search to escape from the local optima.

* For No-Group, the sum of *𝑟𝑁*1 and *𝑟𝑁*2 exceeds 45% on 12 out of

24 instances and it is below 50% on all the instances. However, it

exceeds 50% on 19 out of 24 instances for GATS-PR. Because GATS- PR obtains better solutions than No-Group, it indicates that when the number of move types is large, grouping multiple move types into one neighborhood intensifies the search and thus enhances the search capability.

# 6. Conclusion and future research

In this paper, we propose the GATS-PR algorithm which hybridizes granular neighborhoods, adaptive neighborhood selection and solution- based tabu search with path relinking to solve the MDOVRP. This is the first time that path relinking is used for the MDOVRP to the best of our knowledge and a new similarity definition is designed for this prob- lem. We develop the adaptive neighborhood selection strategy to select suitable move types from a total of 22 ones and exclude unpromising neighborhood solutions by the granular neighborhoods. By forbidding accessing visited solutions, the solution-based tabu search can further enhance diversification of the search.

GATS-PR improved and matched the previous best known results on 4 and 19 out of 24 instances, respectively. By comparing the results obtained by GATS-PR with those of the state-of-the-art algorithms, we can observe that GATS-PR outperforms them for solving the MDOVRP. Importance analysis of the proposed algorithmic components demon- strates that granular neighborhoods, adaptive neighborhood selection,

solution-based tabu search and path relinking all play essential roles in obtaining high-quality solutions eﬃciently.

In future research, we intend to apply GATS-PR to other VRP-related problems, such as the multi-depot VRP and the multi-depot green VRP. The advanced population management mechanism is another topic that deserves to be investigated. We are also going to design other similarity definitions between solutions and integrate them to improve the search capability of path relinking to solve the MDOVRP.

# CRediT authorship contribution statement

**Wenhan Shao:** Investigation, Conceptualization, Methodology, Data collection, Software, Validation, Visualization, Writing – origi- nal draft, Writing – review and editing. **Tuanyue Xiao:** Investigation, Methodology, Data collection, Software, Validation, Writing – original draft, Writing – review and editing. **Zhouxing Su:** Conceptualization, Methodology, Writing – review and editing, Funding acquisition. **Jun- wen Ding:** Conceptualization, Methodology, Writing – review and edit- ing, Supervision, Project administration, Funding acquisition. **Zhipeng Lü:** Conceptualization, Methodology, Writing – review and editing, Funding acquisition.

# Declaration of competing interest

The authors declare that there is no competing interest.

# Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants 72101094 and 62202192, and the Special Project for Knowledge Innovation of Hubei Province under Grant 2022013301015175.









**Fig. 6.** Statistics on the ratios of the neighborhood selection of No-ANS, No-Group and GATS-PR.

# References

1. [Dantzig GB, Ramser JH. The truck dispatching problem. Manag Sci](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib63C906DFFFC619EFF50FE270C4251B03s1) [1959;6(1):80–91.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib63C906DFFFC619EFF50FE270C4251B03s1)
2. [Xidias E, Zacharia P, Nearchou A. Intelligent fleet management of autonomous ve-](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibA81C35C7EADEA0EE5D64C2BE01C887A8s1) [hicles for city logistics. Appl Intell 2022;52(15):18030–48.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibA81C35C7EADEA0EE5D64C2BE01C887A8s1)
3. [Zou G, Tang J, Yilmaz L, Kong X. Online food ordering delivery strategies based on](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib65DD1277C0926F2E5A568FD2ED80BAE9s1) [deep reinforcement learning. Appl Intell 2022;52(6):6853–65.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib65DD1277C0926F2E5A568FD2ED80BAE9s1)
4. [Alitappeh RJ, Jeddisaravi K. Multi-robot exploration in task allocation problem.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibCD319C6615B9C714617C1223DF2F6AB4s1) [Appl Intell 2022;52(2):2189–211.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibCD319C6615B9C714617C1223DF2F6AB4s1)
5. [Kaabi J, Harrath Y, Mahjoub A, Hewahi N, Abdulsattar K. A 2-phase approach for](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib3EA560425539BBDEC8035BD235AA20B8s1) [planning of hazardous waste collection using an unmanned aerial vehicle. 4OR Nov.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib3EA560425539BBDEC8035BD235AA20B8s1) [2022.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib3EA560425539BBDEC8035BD235AA20B8s1)
6. [Vidal T, Crainic TG, Gendreau M, Lahrichi N, Rei W. A hybrid genetic algorithm for](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibB1509EA4917D5C5A82DB0C71877BA4DAs1) [multidepot and periodic vehicle routing problems. Oper Res 2012;60(3):611–24.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibB1509EA4917D5C5A82DB0C71877BA4DAs1)
7. [Tarantilis CD, Kiranoudis CT. Distribution of fresh meat. J Food Eng](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib42D3C78A653F5455341AB8F8D0CC1843s1) [2002;51(1):85–91.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib42D3C78A653F5455341AB8F8D0CC1843s1)
8. [Liu R, Jiang Z, Geng N. A hybrid genetic algorithm for the multi-depot open vehicle](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib85DDACBCF900BAE9702D1A72DA4DF32Es1) [routing problem. OR Spektrum 2014;36(2):401–21.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib85DDACBCF900BAE9702D1A72DA4DF32Es1)
9. [Garey MR, Johnson DS. Computers and intractability: a guide to the theory of NP-](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibA51157251957561C8EE5EBC6ED526FFCs1) [completeness. USA: W. H. Freeman & Co.; 1990.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibA51157251957561C8EE5EBC6ED526FFCs1)
10. [Granada M, Toro EM, Gallego R. An MIP formulation for the open location-routing](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib89349E779E93537795C70C71DF9961EDs1) [problem considering the topological characteristic of the solution-paths. Networks](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib89349E779E93537795C70C71DF9961EDs1) [2019;74(4):374–88.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib89349E779E93537795C70C71DF9961EDs1)
11. [Lalla-Ruiz E, Mes M. Mathematical formulations and improvements for the multi-](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib2C53F6DEFE5EB7BB6502D87952F27A75s1) [depot open vehicle routing problem. Optim Lett 2021;15(1):271–86.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib2C53F6DEFE5EB7BB6502D87952F27A75s1)
12. [Nucamendi-Guillén S, Padilla AG, Olivares-Benitez E, Moreno-Vega JM. The multi-](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib4ADF1F85B1DFE2F02313A2A8070601F1s1) [depot open location routing problem with a heterogeneous fixed fleet. Expert Syst](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib4ADF1F85B1DFE2F02313A2A8070601F1s1) [Appl 2021;165:113846.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib4ADF1F85B1DFE2F02313A2A8070601F1s1)
13. [Lambert M, Hassani R. Diesel genset optimization in remote microgrids. Appl Energy](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibAAF5C7597B9B092B036425534B932D56s1) [2023;340:121036.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibAAF5C7597B9B092B036425534B932D56s1)
14. [Xiong Z, Zhao M, Tan L, Cai L. Real-time power optimization for application](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibCE020EC7E366EF9F46DE603C4D304D38s1) [server clusters based on mixed-integer programming. Future Gener Comput Syst](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibCE020EC7E366EF9F46DE603C4D304D38s1) [2022;137:260–73.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibCE020EC7E366EF9F46DE603C4D304D38s1)
15. [Vardhan BVS, Khedkar M, Srivastava I. Effective energy management and cost effec-](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibB9A647A1FA2A0BF6C21B4A150B722472s1) [tive day ahead scheduling for distribution system with dynamic market participants.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibB9A647A1FA2A0BF6C21B4A150B722472s1) [Sustain Energy Grids Netw 2022;31:100706.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibB9A647A1FA2A0BF6C21B4A150B722472s1)
16. [Lalla-Ruiz E, Expósito-Izquierdo C, Taheripour S, Voß S. An improved formulation](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib3F9A76BEDFCCCA83AF3F0E2C0B6D227As1) [for the multi-depot open vehicle routing problem. OR Spektrum 2016;38(1):175–87.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib3F9A76BEDFCCCA83AF3F0E2C0B6D227As1)
17. [Wang H, Li R, Gong W. Minimizing tardiness and makespan for distributed het-](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib3612A603A6F34A6F1C150E58FAF5DFC0s1) [erogeneous unrelated parallel machine scheduling by knowledge and Pareto-based](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib3612A603A6F34A6F1C150E58FAF5DFC0s1) [memetic algorithm. Egypt Inform J 2023;24(3):100383.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib3612A603A6F34A6F1C150E58FAF5DFC0s1)
18. [Cui Z, Li B, Lan Z, Xu Y. Many-objective evolutionary algorithm based on three-way](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib401A24EC0C993DDFC64961D6BF908199s1) [decision. Egypt Inform J 2023;24(3):100388.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib401A24EC0C993DDFC64961D6BF908199s1)
19. [Zakaryia SA, Ahmed SA, Hussein MK. Evolutionary oﬄoading in an edge environ-](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibAB9EF3E2D37809C77109AE917787B72Es1) [ment. Egypt Inform J 2021;22(3):257–67.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibAB9EF3E2D37809C77109AE917787B72Es1)
20. [Utamima A. A comparative study of hybrid estimation distribution algorithms in](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibB469A108272FA6F3103113C0C9D5B46Bs1) [solving the facility layout problem. Egypt Inform J 2021;22(4):505–13.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibB469A108272FA6F3103113C0C9D5B46Bs1)
21. [Lin A, Li S, Liu R. Mutual learning differential particle swarm optimization. Egypt](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibA4C4737E7236AF10AF6DBE90048C1C14s1) [Inform J 2022;23(3):469–81.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibA4C4737E7236AF10AF6DBE90048C1C14s1)
22. [Chen W-N, Tan D-Z. Set-based discrete particle swarm optimization and its applica-](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibF0188D2262D592A15741A0F223D080BFs1) [tions: a survey. Front Comput Sci 2018;12(2):203–16.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibF0188D2262D592A15741A0F223D080BFs1)
23. [Yang W, Ke L. An improved fireworks algorithm for the capacitated vehicle routing](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib56AC5E5579416B64B900055B75E5CC57s1) [problem. Front Comput Sci 2019;13(3):552–64.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib56AC5E5579416B64B900055B75E5CC57s1)
24. [Liu G, Guo W, Li R, Niu Y, Chen G. XGRouter: high-quality global router in X-](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib60347CD7E04361CA06ADE050F14D9CF2s1) [architecture with particle swarm optimization. Front Comput Sci 2015;9(4):576–94.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib60347CD7E04361CA06ADE050F14D9CF2s1)
25. [Liu S, Huang W, Ma H. An effective genetic algorithm for the fleet size and mix](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib2AE263453C1CF604421A7C05A8D15361s1) [vehicle routing problems. Transp Res, Part E, Logist Transp Rev 2009;45(3):434–45.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib2AE263453C1CF604421A7C05A8D15361s1)
26. [Soto M, Sevaux M, Rossi A, Reinholz A. Multiple neighborhood search, tabu search](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibCD78FEBCD0C4AE8358BD339AD66B536Cs1) [and ejection chains for the multi-depot open vehicle routing problem. Comput Ind](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibCD78FEBCD0C4AE8358BD339AD66B536Cs1) [Eng 2017;107:211–22.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibCD78FEBCD0C4AE8358BD339AD66B536Cs1)
27. [Brandão J. A memory-based iterated local search algorithm for the multi-depot open](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibA2E281A9D8C51566BE226DBB8019EFCEs1) [vehicle routing problem. Eur J Oper Res 2020;284(2):559–71.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibA2E281A9D8C51566BE226DBB8019EFCEs1)
28. [Sadati MEH, Çatay B, Aksen D. An eﬃcient variable neighborhood search with](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib42D7633C7DC7CBB16091DE94EC31835As1) [tabu shaking for a class of multi-depot vehicle routing problems. Comput Oper Res](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib42D7633C7DC7CBB16091DE94EC31835As1) [2021;133:105269.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib42D7633C7DC7CBB16091DE94EC31835As1)
29. [Abdzadeh B, Noori S, Ghannadpour SF. Simultaneous scheduling of multiple con-](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib761E62EB8662AE01EA82837E22CA2470s1) [struction projects considering supplier selection and material transportation routing.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib761E62EB8662AE01EA82837E22CA2470s1) [Autom Constr 2022;140:104336.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib761E62EB8662AE01EA82837E22CA2470s1)
30. [Toth P, Vigo D. The granular tabu search and its application to the vehicle-routing](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibF461F8FD59C426EC064D1A4894F6DF06s1) [problem. INFORMS J Comput 2003;15(4):333–46.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibF461F8FD59C426EC064D1A4894F6DF06s1)
31. [Glover F. Future paths for integer programming and links to artificial intelligence.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibA59AF492496ECEC364683BC3C581CD2Ds1) [Comput Oper Res 1986;13(5):533–49.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibA59AF492496ECEC364683BC3C581CD2Ds1)
32. [Escobar JW, Linfati R, Baldoquin MG, Toth P. A granular variable tabu neigh-](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib65FE3AB744D45B67E1B5C01CC87D180Fs1) [borhood search for the capacitated location-routing problem. Transp Res, Part B,](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib65FE3AB744D45B67E1B5C01CC87D180Fs1) [Methodol 2014;67:344–56.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib65FE3AB744D45B67E1B5C01CC87D180Fs1)
33. [Schneider M, Schwahn F, Vigo D. Designing granular solution methods for routing](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibCE74017613721870F6854B9D60623B50s1) [problems with time windows. Eur J Oper Res 2017;263(2):493–509.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibCE74017613721870F6854B9D60623B50s1)
34. [Shahmanzari M, Aksen D. A multi-start granular skewed variable neighborhood tabu](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib504F08E2C41384368DDA7F408695FA66s1) [search for the roaming salesman problem. Appl Soft Comput 2021;102:107024.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib504F08E2C41384368DDA7F408695FA66s1)
35. [Máximo VR, Nascimento MCV. A hybrid adaptive iterated local search with di-](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib318B478C50301A11D34F7300501F0131s1) [versification control to the capacitated vehicle routing problem. Eur J Oper Res](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib318B478C50301A11D34F7300501F0131s1) [2021;294(3):1108–19.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib318B478C50301A11D34F7300501F0131s1)
36. [Sörensen K, Schittekat P. Statistical analysis of distance-based path relinking for the](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibFEE9D046B5A7A1B7497543B5E601DF47s1) [capacitated vehicle routing problem. Comput Oper Res 2013;40(12):3197–205.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bibFEE9D046B5A7A1B7497543B5E601DF47s1)
37. [Schneider M, Löﬄer M. Large composite neighborhoods for the capacitated location-](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib70DAF27D34160264CFE7B8CD294222B2s1) [routing problem. Transp Sci 2019;53(1):301–18.](http://refhub.elsevier.com/S1110-8665(23)00076-2/bib70DAF27D34160264CFE7B8CD294222B2s1)